

OHJEISTUS SOVELLUSVIRTUALISOINTIIN

Jaakko Paasi

Opinnäytetyö
Toukokuu 2011

Tietotekniikka
Tekniikan ja liikenteen ala



JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES



Tekijä(t) PAASI, Jaakko	Julkaisun laji Opinnäytetyö	Päivämäärä 04.05.2011
	Sivumäärä 87	Julkaisun kieli Suomi
	Luottamuksellisuus () saakka	Verkojulkaisulupa myönnetty (X)
Työn nimi OHJEISTUS SOVELLUSVIRTUALISOINTIIN		
Koulutusohjelma Tietotekniikka		
Työn ohjaaja(t) RANTONEN, Mika		
Toimeksiantaja(t) HUS-Tietotekniikka liikelaitos		
<p>Tiivistelmä</p> <p>Opinnäytetyö tehtiin HUS-Tietotekniikka liikelaitokselle. Työssä tutkittiin sovellusvirtualisointia ja kahta toisistaan poikkeavaa sovellusvirtualisointitekniikkaa. Tutkimuksen tarkoituksena oli selvittää, miten sovellusten virtualisointi ja niiden käyttö toteutetaan valituilla tekniikoilla sekä kuinka toimivia ratkaisuja ne ovat.</p> <p>Opinnäytetyön tavoitteena oli laatia selvän kokonaiskuvan antava dokumentti sovellusvirtualisoinnista ja Microsoft App-V- sekä VMware Thinapp -sovellusvirtualisointitekniikoista. Dokumentin tuli olla laajuudeltaan niin kattava, että lukijan ei tarvitse selata lukuisia alan teoksia ja valmistajien omia dokumentteja saadakseen kokonaisvaltaisen käsityksen aiheesta.</p> <p>Tutkimuksessa testattiin vain toimeksiantajan käytössä olevien sovellusten virtualisointia. Sovellukset virtualisointiin useita kertoja erilaisia virtualisointitekniikoiden tarjoamia asetuksia käyttäen. Testaus toteutettiin suoratoistamalla sovelluksia verkon yli sekä paikallisilla asennuksilla työasemilla.</p> <p>Tutkimuksessa onnistuttiin selvittämään, kuinka App-V:llä ja ThinAppilla sovellusten virtualisointi ja niiden käyttö voidaan toteuttaa, mitä pitää ottaa huomioon ja miten valittujen sovellusten virtualisointi toteutettiin. Työstä saatuja tuloksia hyödyntämällä työn toimeksiantaja voi tehdä vaivattomammin sovellusvirtualisointiin liittyviä päätöksiä.</p>		
Avainsanat (asiasanat) Sovellusvirtualisointi, suoratoisto, ThinApp, App-V		
Muut tiedot		



Author(s) PAASI, Jaakko	Type of publication Bachelor's Thesis	Date 04.05.2011
	Pages 87	Language Finnish
	Confidential () Until	Permission for web publication (X)
Title GUIDANCE TO APPLICATION VIRTUALIZATION		
Degree Programme Information Technology		
Tutor(s) RANTONEN, Mika		
Assigned by HUS-Tietotekniikka		
<p>Abstract</p> <p>The assignment for this thesis came from HUS-Tietotekniikka. This thesis discusses application virtualization and two divergent application virtualization techniques. The study strives for investigating, how the virtualization of applications can be implemented with the selected techniques. The aim was to examine the functionality of the applications.</p> <p>The objective of the thesis was to produce a document which covers application virtualization and two application virtualization techniques. The techniques applied were VMware ThinApp and Microsoft App-V. The produced document had to be extensive enough that the reader does not have to examine other literature of the field to gain a comprehensive picture of the subject.</p> <p>The study was narrowed to cover the applications used by the assigning company. The applications were virtualized various times using divergent configurations given by virtualization techniques. Testing was implemented both by streaming applications over the network and local installations in workstations.</p> <p>As a result of the thesis it was determined, how the virtualization of applications can be implemented and applied by App-V and ThinApp, what has to be considered and how the virtualization of the applications succeeded. By utilizing the results of this study the assigning company is able to make decisions concerning application virtualization with ease.</p>		
Keywords Application virtualization, streaming, App-V, ThinApp		
Miscellaneous		

SISÄLTÖ

LYHENTEET	4
1 TYÖN LÄHTÖKOHDAT	5
1.1 Toimeksiantaja	5
1.2 Tausta	6
1.3 Tehtävä ja tavoitteet	6
2 VIRTUALISOINTI	7
3 SOVELLUSVIRTUALISOINTI	8
3.1 Yleistä	8
3.2 Sovellusvirtualisoinnin hyödyt	11
3.2.1 Yleisesti	11
3.2.2 Sovellusten käyttäytyminen	11
3.2.3 Sovellusten jakelu ja hallinta	13
3.2.4 Paketointi	14
3.3 Heikkoudet	15
4 SOVELLUSVIRTUALISOINTIRATKAISUT	16
4.1 Microsoft App-V	16
4.1.1 Yleistä	16
4.1.2 Komponentit	16
4.1.3 Sovelluksen virtualisoinnin vaiheet	18
4.1.4 App-V:n toimintamallit	19
4.1.5 Rajoitukset	24
4.1.6 Sovellusten linkitys	24
4.2 VMware ThinApp	25
4.2.1 Yleistä	25
4.2.2 Komponentit	26
4.2.3 Sovelluksen virtualisoinnin vaiheet	26
4.2.4 ThinAppin toimintamallit	29
4.2.5 Rajoitukset	30
4.2.6 Sovellusten linkitys	30

5 TESTAUS.....	31
5.1 Testiympäristö.....	31
5.2 Virtualisoitavat sovellukset ja niiden testaus.....	32
5.3 App-V	33
5.3.1 Sekvensserin käyttö	33
5.3.2 Täyden infrastruktuurin malli.....	40
5.3.3 Itsenäinen malli	47
5.3.4 Sovellusten linkitys.....	49
5.4 ThinApp	50
5.4.1 Setup Capturen käyttö.....	50
5.4.2 Streamed Execution Mode	58
5.4.3 Deployed Execution Mode.....	59
5.4.4 Sovellusten linkitys.....	60
6 TESTAUKSEN TULOKSET	60
6.1 App-V	60
6.2 ThinApp	62
7 POHDINTA.....	63
LÄHTEET	66
LIITTEET.....	68
Liite 1. IIS:n asennus.....	68
Liite 2. SQL-palvelimen asennus.....	70
Liite 3. Microsoft Application Virtualization Management Serverin asennus.....	75
Liite 4. Asiakasohjelman asennus.....	82

KUVIOT

KUVIO 1. HUS-Tietotekniikka liikelaitoksen organisaatio	5
KUVIO 2. Perinteinen ja virtualisoitu ympäristö	9
KUVIO 3. Perinteisten sovellusten käyttäytyminen	12
KUVIO 4. Virtualisoidun sovelluksen käyttäytyminen	13
KUVIO 5. Itsenäinen malli.....	20
KUVIO 6. Suoratoistomalli.....	21

KUVIO 7. Täyden infrastruktuurin malli.....	22
KUVIO 8. SCCM & App-V yhdessä.....	23
KUVIO 9. Paketin luonti, muokkaus tai päivitys	34
KUVIO 10. Paketin nimeäminen.....	35
KUVIO 11. Monitorointi.....	36
KUVIO 12. Monitoroinnissa havaitut sovellukset.....	37
KUVIO 13. FB1-lohkon muodostus	38
KUVIO 14. Paketin tiedot	39
KUVIO 15. Sovelluksen jakeluun liittyviä asetuksia	40
KUVIO 16. Hallintapalvelimen käyttöliittymä	41
KUVIO 17. Virtualisoidun sovelluksen asetukset.....	42
KUVIO 18. OSD-tiedoston osittainen sisältö.....	43
KUVIO 19. Sovelluksen lisääminen hallintapalvelimelle	44
KUVIO 20. Pikakuvakkeiden sijainnin määrittäminen	45
KUVIO 21. Yhteenveto sovelluksen lisäämisestä.....	46
KUVIO 22. Näkymä sovellusten lisäyksen jälkeen.....	46
KUVIO 23. Paketin relatiivisen polun tarkastus	47
KUVIO 24. Virtualisoidun sovelluksen asennus MSI-paketilla.....	48
KUVIO 25. MSI-paketilla asennetun virtuaalisovelluksen poistaminen	49
KUVIO 26. Linkityksen muodostus	50
KUVIO 27. Setup Capture-prosessi kokonaisuudessaan	51
KUVIO 28. Järjestelmän skannaus vertailukohdan muodostamiseksi	52
KUVIO 29. Sovelluksen asennus & postscan	53
KUVIO 30. Aloitus pisteiden valinta.....	54
KUVIO 31. Käyttöoikeuksien määrittäminen	55
KUVIO 32. Eristystilan määrittäminen.....	56
KUVIO 33. Hiekkalaatikon sijainnin määrittäminen	57
KUVIO 34. Paketoinnin viimeinen vaihe	58
KUVIO 35. Pikakuvakkeen polku	59

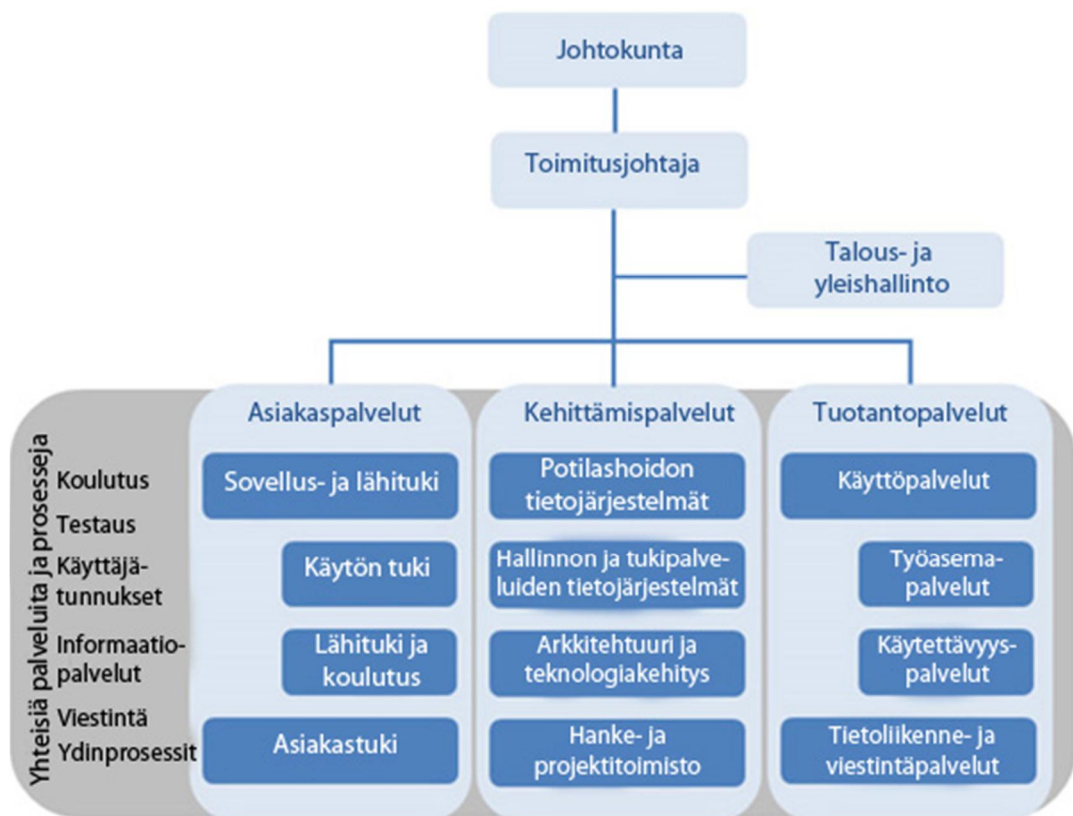
LYHENTEET

COM	Component Object Model
DCOM	Distributed Component Object Model
DLL	Dynamic Link Library
ESD	Electronic Software Delivery
FB	Feature Block
HUS	Helsingin ja Uudenmaan sairaanhoitopiiri
IIS	Internet Information Services
MMC	Microsoft Management Console
MSI	Microsoft Installer
NTFS	New Technology File System
RTSP	Real Time Streaming Protocol
HTTP	Hypertext Transfer Protocol
SCCM	System Center Configuration Manager
SLA	Service Level Agreement
SMB	Server Message Block
UNC	Universal Naming Convention
XML	Extensible Markup Language

1 TYÖN LÄHTÖKOHDAT

1.1 Toimeksiantaja

HUS-Tietotekniikka on liikelaitos, joka toimittaa tietojärjestelmä- ja tietotekniikka-palveluita HUS-kuntayhtymälle, kuntayhtymän yhteistyökumppaneille, liikelaitoksille ja tytär- ja osakkuusyhteisöille. Muina asiakkaina ja sidosryhminä ovat kunnat, yliopistot, yksityiset terveydenhuollon toimijat ja viranomaiset. HUS-Tietotekniikassa työskentelee noin 165 henkilöä kehitys-, käyttö-, ylläpito- ja tukipalveluiden parissa. HUS-Tietotekniikan organisaatiokaavio on nähtävissä kuvioista 1. HUS-Tietotekniikka ylläpitää noin 16 000 työasemaa, joista suurin osa käyttää Windows XP -käyttöjärjestelmää. (HUS-Tietotekniikka 2011.)



KUVIO 1. HUS-Tietotekniikka liikelaitoksen organisaatio (HUS-Tietotekniikka 2011, muokattu)

1.2 Tausta

Microsoftin kehittämä Windows 7 -käyttöjärjestelmä on kasvattanut suosiotaan organisaatioiden keskuudessa, sillä lukuisat organisaatiot ovat siirtymässä siihen nopealla aikataululla. Windows 7 nähdään varteenotettavana vaihtoehtona Windows XP -käyttöjärjestelmän seuraajaksi, jota suurin osa organisaatioista käyttää. (Brodkin 2010.)

Työn toimeksiantaja on päättänyt siirtyä käyttämään Windows 7 -käyttöjärjestelmää. Uuteen käyttöjärjestelmään siirtyminen on usein loppukäyttäjän toimintaa häiritsevää, kallis ja aikaa vievä prosessi. Kustannuksia aiheuttaa esimerkiksi regressiotestaus, joka joudutaan suorittamaan uudestaan sovellusten toiminnan varmistamiseksi. Lisäksi ongelmaksi muodostuvat sovellukset, joille tukea ei ole vielä tarjolla. Osalle sovelluksista toimiva versio on kehitteillä, mutta löytyy myös sellaisia sovelluksia, joille tukea ei tulla tarjoamaan. Tällaisia sovelluksia ovat muun muassa legacy-sovellukset eli sovellukset, jotka on kehitetty aikanaan tietylle alustalle..

Sovellusvirtualisointia markkinoidaan yllä mainittujen ongelmien ratkaisijana. Sen avulla saadaan vähennettyä resursseja vievää regressiotestausta sekä mahdollistetaan vanhojen sovellusten toimiminen uusissakin käyttöjärjestelmissä. Tästä johtuen sovellusvirtualisointia ja sen tuomia mahdollisuuksia lähdettiin tutkimaan toimeksiantajan töiden helpottamiseksi.

1.3 Tehtävä ja tavoitteet

Työn lähtökohtana oli tutkia sovellusvirtualisointia ja kahta toisistaan poikkeavaa sovellusvirtualisointitekniikkaa. Tutkimuksella pyrittiin selvittämään, miten sovellusten virtualisointi ja niiden käyttö toteutetaan valituilla tekniikoilla, sekä kuinka toimivia ratkaisuja ne ovat. Tutkimusstrategiana käytettiin tapaustutkimusta. Tapaustutkimuksessa valitaan yksittäinen tapaus tai pieni joukko tapauksia. Valitusta tapauksesta tai tapauksista tuotetaan yksityiskohtaista ja intensiivistä tietoa. (Hirsjärvi, Remes & Sa-

javaara 2009, 135.) Tapaustutkimus valittiin tutkimusstrategiaksi, koska työn tarkoituksena oli tuottaa yksityiskohtaista tietoa sovellusvirtualisointitekniikoista.

Tutkimuksen pohjalta laadittiin dokumentti, joka kattoi VMware ThinApp- ja Microsoft App-V-tekniikoiden ominaisuudet, niiden vaatimat asennukset sekä valittujen sovellusten virtualisoinnin ja testauksen rakennetussa testiympäristössä. Tavoitteena oli laatia selvän kokonaiskuvan antava dokumentti sovellusvirtualisoinnista, sen tuomista mahdollisuuksista ja ennen kaikkea valituista sovellusvirtualisointitekniikoista. Dokumentin tuli olla laajuudeltaan niin kattava, että lukijan ei tarvitse selata lukuisia alan teoksia ja valmistajien omia dokumentteja saadakseen kokonaisvaltaisen käsityksen aiheesta.

Työn yksi tärkeimmistä tavoitteista oli onnistua sovellusten virtualisoinnissa siten, että ne toimisivat käyttöönotettavassa Windows 7 -käyttöjärjestelmässä. Tällöin sovellusvirtualisointia voitaisiin hyödyntää myöhemmin muun muassa uuteen käyttöjärjestelmään siirryttäessä. Työssä keskityttiin virtualisoimaan vain toimeksiantajan käytössä olevia sovelluksia.

Testausvaiheella pyrittiin selvittämään sovellusvirtualisointitekniikoiden käyttöönoton helppoutta, hallittavuutta ja tekniikoilla virtualisoitujen sovellusten toimivuutta. Testauksessa saatuja tuloksia on tarkoitus hyödyntää jatkossa, kun toimeksiantaja tekee sovellusvirtualisointiin liittyviä päätöksiä.

2 VIRTUALISOINTI

Virtualisointi on määritelty monin eri tavoin. Esimerkiksi Yleisen suomalaisen asiasanaston mukaan virtualisoinnilla tarkoitetaan tietojenkäsittelyssä tekniikkaa, jolla jonkin fyysisen resurssin tekniset piirteet piilotetaan muilta resursseja käyttäviltä järjestelmiltä, sovelluksilta ja loppukäyttäjiltä. Spruijtin (2010, 15) mukaan virtualisointi on prosessi, jossa käytettävät resurssit eristetään toisistaan.

IBM hyödynsi virtualisointia jo yli 30 vuotta sitten osioimalla suuria keskustietokoneita. Luoduista osioista tehtiin erillisiä virtuaalikoneita, jotka mahdollistivat useiden ohjelmien ja prosessien suorittamisen yhtä aikaa. Näin kalliiden keskustietokoneiden tarjoamille resursseille saatiin parempi käyttöaste. (VMware 2011a.)

Virtualisointi on kehittynyt vuosien varrella ja se on nykyään varsin laaja käsite. Se voidaan jakaa lukuisiin eri osa-alueisiin (Ruest & Ruest 2009, 25–26):

- palvelinvirtualisointi (Server Virtualization)
- ohjelmistovirtualisointi (Software Virtualization)
- laitteistovirtualisointi (Hardware Virtualization)
- tietovarastovirtualisointi (Storage Virtualization)
- verkon virtualisointi (Network Virtualization)
- työpöytävirtualisointi (Desktop Virtualization)
- sovellusvirtualisointi (Application Virtualization).

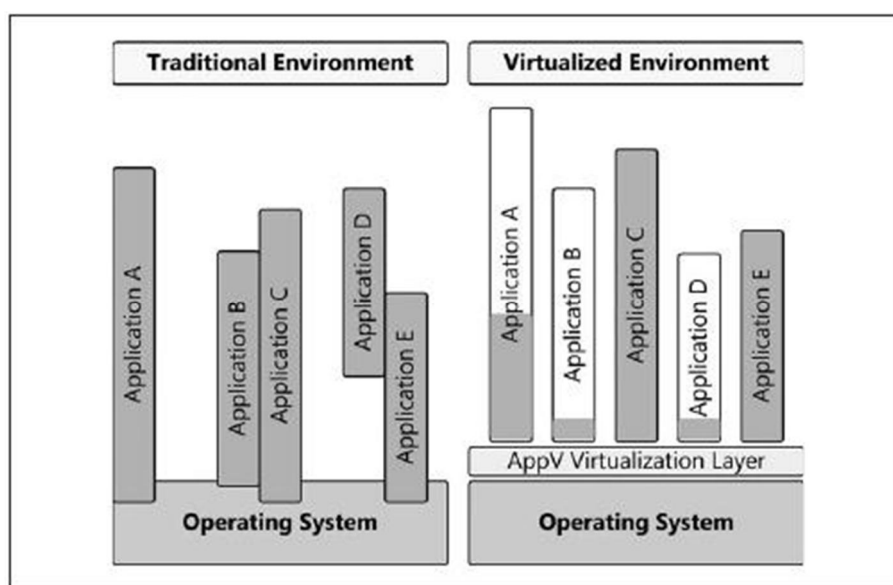
3 SOVELLUSVIRTUALISOINTI

3.1 Yleistä

Sovellusvirtualisointi on yksi tämän hetken kiinnostavimmista virtualisointiin liittyvistä tekniikoista. Se esiteltiin jo vuosia sitten, mutta se on alkanut yleistyä vasta viime vuosina. Yleistymistä on hidastanut muun muassa vähäinen tieto sovellusvirtualisoinnista. Monet organisaatiot totuttelevat vasta muihin virtualisointimentelmiin kuten palvelinvirtualisointiin. Lisäksi päätöksentekijät organisaatioissa eivät pidä tietotekniikkaa investointina vaan välttämättömänä kuluna, joten siihen ei haluta investoida. Uutta tekniikkaa kohtaan löytyy usein myös ennakkoluuloja: sen pelätään esimerkiksi tuovan lisää monimutkaisia vaiheita tai lisäävän kustannuksia. (Alvarez 2011, 7-9.)

Ajat ovat kuitenkin muuttuneet ja yritykset ovat viimein ymmärtäneet sovellusvirtualisoinnin tuomat hyödyt. Se nähdään varteenotettavana vaihtoehtona myös sovellusten jakelussa. (Spruijt 2010, 17.) Yksi syy yleistymiseen on myös uusi Windows 7-käyttöjärjestelmä, johon monet organisaatiot ovat siirtymässä. Sovellusvirtualisointitekniikoiden toimittajat, kuten VMware, esittelevät kotisivuillaan sovellusvirtualisointia ratkaisuna, joka auttaa uuteen käyttöjärjestelmään siirtymisessä. (VMware 2011b.) Monet organisaatiot ovat jo siirtyneet Windows 7 -käyttöjärjestelmään sovellusvirtualisointia hyödyntäen. Esimerkiksi Oulun yliopisto hyödynsi sovellusvirtualisointia helpottaakseen siirtymistä uuteen käyttöjärjestelmään. (Oksanen 2010.)

Sovellusvirtualisoinnilla tarkoitetaan prosessia, jossa sovellus irrotetaan käyttöjärjestelmästä ja muista sovelluksista. Prosessin aikana sovelluksen asetukset, tiedostot, rekisteriavaimet ja kirjastot, toisin sanoen koko toimintaympäristö tallennetaan ja yhdistetään yhdeksi helposti jaettavaksi paketiksi. Sovellus toimii omassa virtuaalikerroksessa, jota kutsutaan myös ”kuplaksi” tai ”hiekkalaatikoksi”. Virtuaalikerros suojaa käyttöjärjestelmää sovelluksen aiheuttamilta muutoksilta. Kuviosta 2 nähdään, kuinka virtuaalinen kerros erottaa sovellukset ja käyttöjärjestelmän toisistaan, kun taas tavallisessa ympäristössä sovellukset tunkeutuvat käyttöjärjestelmän sisään. (Spruijt 2010, 18.)



KUVIO 2. Perinteinen ja virtualisoitu ympäristö (Ruest ym. 2009, 274)

Virtualisoitu sovellus ei tee minkäänlaisia muutoksia käyttöjärjestelmän rekisteriin tai tiedostojärjestelmään. Siitä huolimatta se pystyy keskustelemaan käyttöjärjestelmän ja muiden asennettujen sovellusten kanssa aivan kuten perinteisesti asennettu sovellus. Tämä mahdollistaa saman ohjelman eri versioiden käyttämisen (esimerkiksi Microsoft Word 2003 ja Microsoft Word 2010) samassa käyttöjärjestelmässä. (Ruest ym. 2009, 273.)

Sovellusvirtualisointiprosessi koostuu viidestä eri osasta: nauhoitus (capture), eristys (isolation), linkitys (linking), jakelu (deployment) sekä päivitys (upgrade). Nauhoitusvaiheessa sovellus ja sen käyttämät resurssit paketoidaan uudestaan virtuaalisovellukseksi. Eristyksessä määritellään, kuinka näkyvää sovelluksen toiminta on käyttöjärjestelmälle. Osa sovelluksista tarvitsee käyttöjärjestelmän resursseja toimiakseen, jolloin sitä ei voida eristää kokonaan käyttöjärjestelmästä. Linkityksessä voidaan määrittää sovellukset, joiden kanssa tehdään yhteistyötä. Riippuen loppukäyttäjistä ja sovelluksen ominaisuuksista, virtualisoidun sovelluksen jakelu vaihtelee paikallisesti asennetusta sovelluksesta pelkästään sovelluksen esittämiseen käyttäjälle. Sovelluksen päivitystapa riippuu käytettävästä tekniikasta ja sovelluksen toimitustavasta käyttäjälle. Yksinkertaisimmillaan sovellus päivitetään serverillä, joka jakaa sovellusta muille laitteille. (Easter 2009, 3.)

Sovellusvirtualisointiratkaisut voidaan jakaa agentillisiin ja agentittomiin ratkaisuihin. Agentillisessa ratkaisussa jokaiselle virtualisoituja sovelluksia käyttävälle työasemalle asennetaan ”agentti” eli asiakasohjelmisto. Agentti sisältää toiminnot jokaisen sovelluksen virtuaaliympäristön asennusta ja ylläpitoa varten sekä huolehtii erilaisista tehtävistä, kuten tiedostotyyppien välisistä assosiaatioista ja suoratoiston käyttäytymisestä. Agentittomassa ratkaisussa virtualisoitu sovellus sisältää sisäänrakennetun agentin, jolloin työasemille ei tarvitse asentaa erikseen ohjelmistoja. (Spruijt 2010, 19.)

Molemmilla ratkaisuilla on sekä hyvät että huonot puolensa. Agentillisessa ratkaisussa agentin pitää olla asennettuna työasemaan etukäteen ennen kuin virtualisoituja sovelluksia pystytään käyttämään. Siten agentillinen ratkaisu sisältää yhden ylimääräisen vaiheen agentittomaan vaiheeseen nähden. Agentti kuitenkin lisää tietoturvaa, koska

sovellus ei toimi ilman asennettua agenttia. Agentittomassa ratkaisussa sovellusta voidaan käyttää missä tahansa, olipa kyseessä sitten yrityksessä tai käyttäjän kotona oleva tietokone. Sitä voidaan käyttää suoraan esimerkiksi USB-muistitikulta. Tästä johtuen sovelluksen lisensseihin ja suojausmekanismeihin pitää kiinnittää erityistä huomiota, sillä muutoin kuka tahansa voi käyttää sovellusta päästessään siihen käsiksi. (Ruest ym. 2009, 277-278.)

3.2 Sovellusvirtualisoinnin hyödyt

3.2.1 Yleisesti

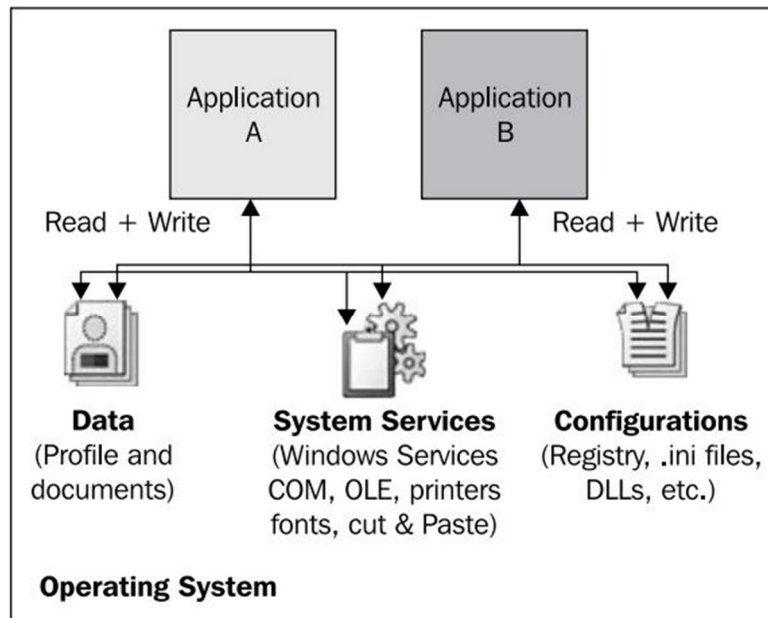
Sovellusvirtualisoinnin selkein hyöty on vanhojen sovellusten toimivuus Windows-käyttöjärjestelmissä, joita sovellus ei tavallisesti tukisi. Tämä mahdollistaa siirtymisen uuteen käyttöjärjestelmään huomattavasti pienemmällä vaivalla. Etuna on myös se, ettei sovelluksia asenneta käyttöjärjestelmään. Tämä ehkäisee yhteensopivuusongelmia sekä ristiriitoja sovellusten välillä. Edellä mainittujen hyötyjen lisäksi sovellusvirtualisointi tarjoaa myös monia muita etuja perinteisiin menetelmiin nähden (Spruijt 2010, 15):

- Sovellusten regressiotestaus käyttää vähemmän resursseja.
- Saman sovelluksen eri versioita voidaan käyttää samassa käyttöjärjestelmässä rinnakkain.
- Sovellusten jakelu nopeutuu ja helpottuu.
- Sovellusten päivittäminen on vaivattomampaa.
- Sovelluksen edelliseen versioon palaaminen helpottuu.

3.2.2 Sovellusten käyttäytyminen

Perinteisesti sovellukset asennetaan suoraan käyttöjärjestelmään. Käyttöjärjestelmää muokataan asennusprosessissa sovelluksen tarpeiden mukaan. Uuden sovelluksen vaatimat asetukset saattavat kirjoittaa olemassa olevien sovellusten asetusten päälle, jolloin ne voivat lakata toimimasta. Kuviosta 3 nähdään, kuinka kummallakin sovel-

luksella on luku- ja kirjoitusoikeus käyttöjärjestelmän konfiguraatioihin, esimerkiksi rekisteriin ja DLL-kirjastoihin. Sovellusten tai jopa käyttöjärjestelmän toiminta voi häiriintyä, jos sovellukset muokkaavat samoja tiedostoja tai rekisteriarvoja. (Alvarez 2011, 12.)

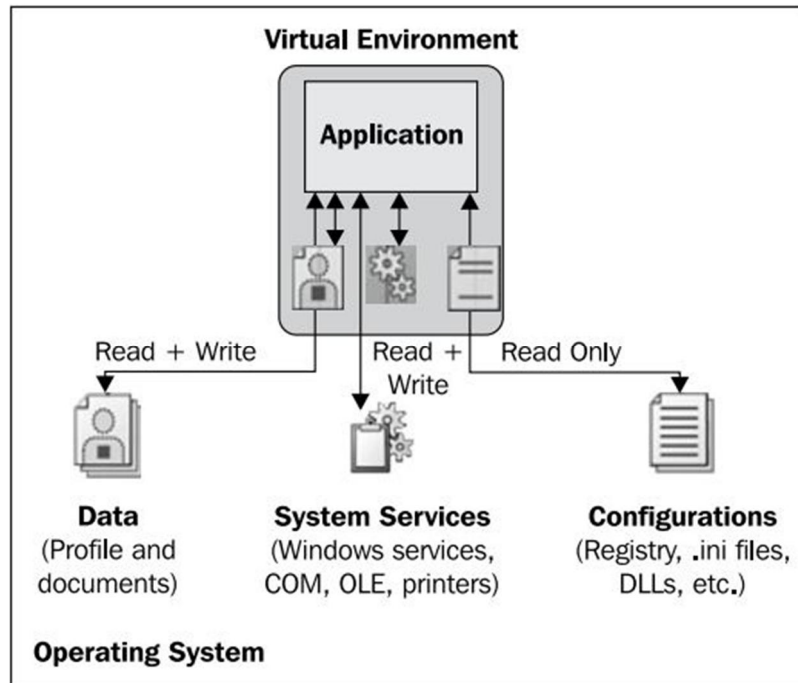


KUVIO 3. Perinteisten sovellusten käyttäytyminen (Alvarez 2011, 12)

Edellä mainituista ongelmista johtuen jokaisen käyttöön tulevan sovelluksen vaikutus käytössä oleviin sovelluksiin pitää testata. Tätä prosessia kutsutaan regressiotestaukseksi. Myös sovelluspäivitysten vaikutukset käyttöjärjestelmään ja sovelluksiin täytyy testata. Siirryttäessä uuteen käyttöjärjestelmään jokainen käytössä oleva sovellus joudutaan testaamaan uudestaan jälleen kerran. Uusien sovellusten regressiotestaus yhteensopivuusongelmien selvittämiseksi on yksi eniten aikaa vievistä toimista, varsinkin käytettäessä useita kymmeniä tai jopa satoja eri sovelluksia. (Ruest ym. 2009, 266.)

Kuviosta 4 nähdään, kuinka virtualisoitu sovellus ei tee muutoksia käyttöjärjestelmään lukuun ottamatta käyttäjäprofiilia ja siellä olevia tiedostoja, joten regressiotestauksen vaatimia resursseja saadaan vähennettyä huomattavasti. Sovellus toimii pääosin omas-

sa virtuaaliympäristössään. Sillä on pelkästään lukuoikeus rekisteriin ja muihin konfiguraatioihin, mutta käyttöjärjestelmän paikalliset palvelut, kuten leikkaa- ja liitä-toiminnot tai tulostaminen ovat silti käytettävissä. (Alvarez 2011, 13.)



KUVIO 4. Virtualisoidun sovelluksen käyttäytyminen (Alvarez 2011, 13)

3.2.3 Sovellusten jakelu ja hallinta

Sovellusvirtualisointi tuo käyttäjille joustavuutta sovellusten hallintaan ja jakeluun suoratoiston (streaming) avulla. Sen avulla virtualisoitu sovellus voidaan jakaa pienempiin lohkoihin, joita lähetään loppukäyttäjälle tarpeen mukaan. Sovelluksen käynnistyslohko, joka riittää sovelluksen käynnistymiseen ja perustoimintoihin, on usein alle 10 % sovelluksen viemästä kokonaistilasta. Tällöin se ei juuri kuormita verkkoa. Useimmat sovellusvirtualisointitekniikat tukevat käynnistyslohkon käyttöä. Osa virtualisointitekniikoista tukee myös ennakoivia lohkoja. Suoratoistopalvelin analysoi lähetettäänsä tietoa ja määrittää sen perusteella suosituimmat lohkot, joista tehdään ennakoivia lohkoja. Nämä lähetetään niille käyttäjille, jotka eivät ole vielä ehtineet

käyttää kyseisiä lohkoja. Seurauksena vasteaika lyhenee ja käyttökokemus paranee. (Ruest ym. 2009, 281.)

Käytettäessä suoratoistoa sovelluksia voidaan hallita keskitetysti. Mahdolliset päivitykset tai sovelluksen poistot tehdään yhdessä paikassa muun muassa suoratoisto- tai tiedostopalvelimella. Muutokset välittyvät loppukäyttäjille seuraavalla sisäänkirjautumiskerralla tai asiakasohjelman päivityksellä. (Mts. 296.)

Helpottaakseen sovellusten hallintaa organisaatiot ovat suunnitelleet useita käyttöjärjestelmän levykuvia (image), joihin sisällytetään sovelluksia tarpeiden mukaan. Sisällytetyt sovellukset asentuvat käyttöjärjestelmän mukana, joten kaikkia sovelluksia ei tarvitse jakaa erikseen. Organisaatioilla voi olla käytössään kymmeniä tai jopa satoja levykuvia, joita päivitetään jatkuvasti. (Mts. 294.)

Suoratoiston avulla hallittavien levykuvien määrä saadaan halutessa vähennettyä esimerkiksi yhteen tai kahteen, sillä levykuvan tulee sisältää vain pelkkä käyttöjärjestelmä, virtualisointitekniikan mahdollisesti vaatima agentti ja muut ohjelmat, joita ei voida tai kannata virtualisoida. (Mts. 294.) Kaikki muut halutut sovellukset tuodaan loppukäyttäjän saataville suoratoistoa käyttäen. Sovellukset voidaan halutessa ladata paikallisesti välimuistiin, jolloin ne ovat käytettävissä myös yhteydettömässä tilassa. (Mts. 283.)

3.2.4 Paketointi

Perinteisesti jokainen paketoitu sovellus täytyy käydä läpi uudestaan siirryttäessä uuteen käyttöjärjestelmään. Paketointiprosessi on kallein sovellusten hallintaan liittyvä tehtävä vaihdettaessa käyttöjärjestelmää. Virtualisoidut sovellukset eivät ole sidottuja tiettyyn käyttöjärjestelmään, joten niitä ei tarvitse paketoida uudestaan uuteen Windows-käyttöjärjestelmään siirryttäessä. Pelkästään tämä on tarpeeksi suuri syy sovellusvirtualisoinnin käyttämiseen. (Ruest ym. 2009, 277.)

3.3 Heikkoudet

Sovellusvirtualisointi ei ole toimiva ratkaisu jokaisen sovelluksen kohdalla. Esimerkiksi virtualisoitujen tietokantasovelluksien suorituskyvyn tiedetään laskevan. (Georgieva 2009.) Myös sovellusvirtualisointitekniikoiden välillä on huomattavia eroa; toisten tekniikoiden avulla voi virtualisoida haastaviakin sovelluksia, kun taas toisilla ei voi virtualisoida esimerkiksi 64-bittisiä sovelluksia tai Internet Explorer 6:sta.

Kaikkien sovellusvirtualisointitekniikoiden kannalta ongelmallisia tapauksia ovat sovellukset, jotka vaativat toimiakseen laiteajurit. Ongelma voidaan kiertää asentamalla laiteajuri paikallisesti työasemaan, mutta samalla joudutaan tekemään yksi vaihe enemmän. Myös ne sovellukset, jotka hyödyntävät kernel-tason komponentteja, ovat vaikeita virtualisoinnin kannalta. Näitä sovelluksia ovat mm. virustorjuntaohjelmat ja palomuurit. (Wikipedia 2011.)

Sovellusvirtualisointi on uusi teknologia, joten sen opettelemiseen täytyy käyttää aikaa. Jos sovellusvirtualisoinnin mahdollistamaa suoratoistoa halutaan käyttää, täytyy varmistaa servereiden ja verkon redundanttisuus sekä riittävä kaistanleveys. (Pietroforte 2008.)

Useimpia ohjelmia ei ole kehitetty toimimaan virtualisoidussa ympäristössä, joten sovellustoimittajat saattavat evätä virtualisoitujen sovellusten tuen, koska sovelluksen suorituskyky ja toimivuus voivat muuttua virtualisoinnin yhteydessä. Mahdollisten ongelmien tullessa ilmi joudutaan tutkimaan, johtuivatko ongelmat itse sovelluksesta vai sen virtualisoinnista. (Pietroforte 2008.) Sovellustoimittajat ovat usein vastuussa sovelluksen toiminnallisuudesta laaditun SLA:n (Service Level Agreement) mukaan, ja siksi toimittajat eivät mielellään tue sovellusvirtualisointia. (Bansal 2010.)

4 SOVELLUSVIRTUALISOINTIRATKAISUT

Sovellusvirtualisoinnin kehitys on ollut varsin nopeaa. Erilaisia ratkaisuja löytyy lukuisilta eri toimittajilta. Tekniikoiden toteutustavat voivat vaihdella paljonkin, mutta lopputulos on lähes sama kaikissa. Tunnetuimpia agentillisia tekniikoita ovat Microsoft App-V, Citrix XenApp ja Symantec SVS. Agentittomista ratkaisuista tunnetuimpia ovat VMware ThinApp sekä InstallFree Bridge. (Ruest 2009, 41–42, 293.) Tässä työssä keskitytään toimeksiantajan pyynnöstä Microsoftin ja VMwaren tarjoamiin ratkaisuihin.

4.1 Microsoft App-V

4.1.1 Yleistä

App-V on Microsoftin tarjoama ratkaisu sovellusvirtualisointiin. App-V:n (aiemmin SoftGrid) kehitti alun perin yhtiö nimeltä Softricity. Microsoft osti Softricityn heinäkuussa vuonna 2006 ja on kehittänyt App-V:tä siitä lähtien. (Microsoft 2006.) App-V:n tuorein versio on 4.6, ja se toi mukanaan tuen mm. Windows 7 ja Windows 2008 R2 -käyttöjärjestelmille sekä 64-bittisille asiakasohjelmille. (Microsoft 2010a, 2.)

App-V:llä virtualisoidun sovelluksen toimitus käyttäjälle tapahtuu kahdessa osassa. Ensimmäisessä osassa sovellus julkaistaan niihin työasemiin, joissa asiakasohjelma on asennettuna. Sovelluksen julkaisuun sisältyy pikakuvakkeiden toimitus, tiedostotyyppien väliset assosiaatiot, paketin määrittystiedot sekä content-kansion sijainti. Toisessa osassa paketoitu sovellus toimitetaan käyttäjälle suoratoistamalla. Vaihtoehtoisesti sovellus voidaan toimittaa käyttäjälle MSI-pakettia hyväksi käyttäen, jolloin julkaisu- ja toimitusvaihe yhdistetään yhdeksi kokonaisuudeksi. (Microsoft 2010a, 6.)

4.1.2 Komponentit

App-V infrastruktuuri koostuu seuraavista komponenteista: hallintapalvelin (Management Server), hallintajärjestelmä (Management System), suoratoistopalvelin

(Streaming Server), asiakasohjelma (Virtualization Client) sekä sekvensseri (sequencer).

Hallintapalvelin

Hallintapalvelin seuraa mm. sovellusten käyttöä ja lisenssejä, suorituskykyä sekä niille asetettuja oikeuksia SQL-tietokannan avulla. Hallintapalvelimen hankkimat tiedot tallennetaan SQL-tietokantaan. Hallinta- sekä SQL-palvelin asennetaan usein samaan verkkosegmenttiin ongelmien välttämiseksi. (Alvarez 2011, 24-25.)

Hallintajärjestelmä

Hallintajärjestelmä koostuu hallintakonsolista (Management Console) ja hallintapalvelusta (Management Service). Hallintajärjestelmä muodostaa yhteyden SQL-tietokannan ja MMC:n (Microsoft Management Console) välille luku- ja kirjoituspyyntöjen kontrolloimiseksi. (Mts. 25.)

Suoratoistopalvelin

Suoratoistopalvelin huolehtii ainoastaan sovellusten suoratoistosta asiakasohjelmille. Suoratoisto on mahdollista toteuttaa kolmella eri protokollalla. RTSP/RTSPS (Real Time Streaming Protocol) on App-V:n natiivisti käyttämä protokolla. Sitä käyttämällä voidaan hyödyntää App-V:n aktiivipäivitystä (Active Upgrade), jolla voidaan päivittää tai muokata luotuja sovelluksia reaaliajassa. HTTP/S:n avulla sovellus voidaan suoratoistaa käyttäjälle esimerkiksi IIS:n (Internet Information Services) tai Apachen avulla käyttämällä tunnettuja portteja (esimerkiksi 80 ja 443). Viimeinen vaihtoehto SMB (Server Message Block) mahdollistaa sovelluksen suoratoiston suoraan tiedostopalvelimelta, joten se on yksinkertainen ja helppo toteuttaa. Ainoastaan RTSP/RTSPS vaatii hallinta- tai suoratoistopalvelimen asennuksen suoratoistamisen suorittamiseksi. (Mts. 25-26.)

Asiakasohjelma

Asiakasohjelma asennetaan loppukäyttäjän työasemalle. Asiakasohjelma huolehtii yhteydenpidosta hallintapalvelimeen, suoratoistetun ohjelman lataamisesta välimuistiin sekä julkaisun päivityksestä. Lisäksi se tallentaa virtuaalisovelluksen sisältämät käyttäjäkohtaiset tiedot käyttäjän profiiliin. (Mts. 26.)

Sekvensseri

Sekvensseriä käytetään sovellusten virtualisointiin. Se nauhoittaa sovelluksen asennuksen käyttöjärjestelmään ja luo saatujen tietojen perusteella paketin. Paketti sisältää useita eri tiedostoja (Mts. 28-29):

- OSD-tiedosto sisältää asiakasohjelman tarvitsemat tiedot sovelluksen löytämisestä ja suorittamista varten tarjoamalla linkin SFT-tiedostoon
- SFT-tiedosto sisältää kaikki alkuperäisen sovelluksen käyttämät tiedostot. Tiedostot paketoidaan nauhoitusvaiheessa suoratoistoa tukevaan muotoon. Se koostuu kahdesta osasta. Ensimmäinen osa käynnistyslohko FB1 (Feature Block 1) sisältää sovelluksen eniten käyttämät osat. Toinen osa FB2 (Feature Block 2) sisältää loput jäljelle jääneet osat
- SPRJ-tiedosto on XML-pohjainen tiedosto, joka sisältää listan sekvensserin luomista tiedostoista, rekisteriavaimista ja kansioista
- Manifest-tiedosto on XML-tiedosto, joka sisältää tietoa virtualisoidun tiedoston pikakuvakkeiden sijainnista (työpöytä, käynnistä-valikko ja pikakäynnistys) sekä tiedostotyyppien välisistä assosiaatioista. Manifest-tiedostoa tarvitaan esimerkiksi käytettäessä System Center Configuration Manager -järjestelmää
- ICO-tiedosto sisältää pikakuvakkeet, jotka ohjaavat asiakasohjelman virtualisoituun sovellukseen
- MSI-tiedosto sisältää manifest-, OSD- ja ICO-tiedostojen kopiot. Sen käyttäminen ei ole pakollista.

4.1.3 Sovelluksen virtualisoinnin vaiheet

Sovelluksen virtualisointi tapahtuu sekvensserillä. Aluksi valitaan uuden paketin luonti. Vaihtoehtoisesti voidaan valita myös olemassa olevan paketin muokkaus tai päivitys. Seuraavaksi paketti nimetään ja lisätään mahdollisia kommentteja. Alkutoimenpiteiden jälkeen siirrytään sovelluksen asennuksen monitorointiin. Monitoroinnissa valitaan sovelluksen asennuskansio, jonka jälkeen sovelluksen asennus käynnistetään. Jotkut sovellukset vaativat päätelaitteen uudelleen käynnistämisen asennuksen loppuun suorittamiseksi. Sekvensseri simuloi uudelleen käynnistyksen ja nauhoittaa

käynnistykseen liittyvät prosessit, joten konetta ei tarvitse käynnistää uudestaan. (Microsoft 2010b, 20-24.)

Sekvensseri analysoi sovelluksen asentamia tiedostoja ja listaa asennetut komponentit, kuten pikakuvakkeet, itse sovelluksen ja tiedostotyyppien väliset assosiaatiot. Komponentteja voidaan poistaa tai lisätä sekä muokata tarpeiden mukaan. Seuraavaksi sovellus tai sovellukset käynnistetään toimivuuden tarkastamiseksi. Käynnistetyistä sovelluksista muodostetaan käynnistyslohko FB1. (Mts. 25-26.)

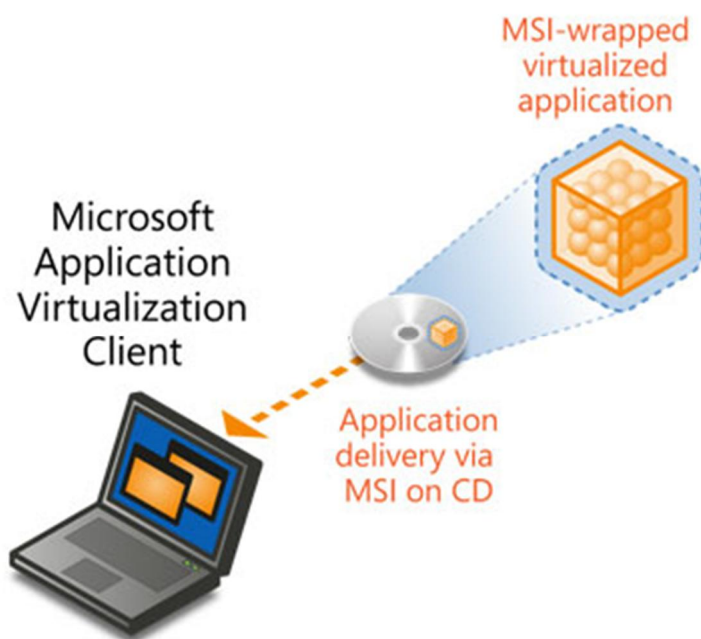
Viimeisenä vaiheena suoritetaan Deployment-välilehdeltä löytyviä sovelluksen jakeluun liittyviä toimenpiteitä. Välilehdeltä valitaan käytettävä protokolla, suoratoistopalvelimen nimi, portti sekä käytettävät käyttöjärjestelmät. Lisäksi voidaan valita, otetaanko turvamäärittelyt (security descriptor) käyttöön, luodaanko MSI-paketti tai pakataanko paketti tilan säästämiseksi. Lopuksi projekti tallennetaan haluttuun paikkaan .sprj-muodossa. (Mts. 27-28.)

4.1.4 App-V:n toimintamallit

App-V tarjoaa erilaisia toimintamalleja yritysten tarpeiden mukaan. Käytettävä malli voi olla itsenäinen malli (Standalone Model), suoratoistomalli (Streaming Model) tai täyden infrastruktuurin malli (Full Infrastructure Model). Lisäksi App-V voidaan integroida SCCM (System Center Configuration Manager) 2007 R2 -järjestelmään, jolloin erillistä järjestelmää ei tarvitse rakentaa.

Itsenäinen malli

App-V vaatii itsenäisessä mallissa toimiakseen pelkästään sekvensserin ja asiakasohjelman. Kuvio 5 havainnollistaa itsenäisen mallin toimintaa. Asiakasohjelma täytyy asettaa itsenäiseen tilaan ennen käyttöä. Sekvensseri luotavalla MSI-paketilla voidaan korvata hallintapalvelimen suorittamia toimintoja. MSI-paketti kopioi SFT-tiedoston välimuistiin, rekisteröi tiedostotyyppien väliset assosiaatiot sekä julkaisee pikakuvakkeet. MSI-paketti ja SFT-tiedosto pitää siirtää käytettävälle työasemalle samaan kansioon ennen asennusta. Siirto voidaan suorittaa melkeinpä millä tahansa keinolla, kuten USB-muistitikulla tai Group Policylla. (Buller ym. 2010, 304.)



KUVIO 5. Itsenäinen malli (Microsoft 2011)

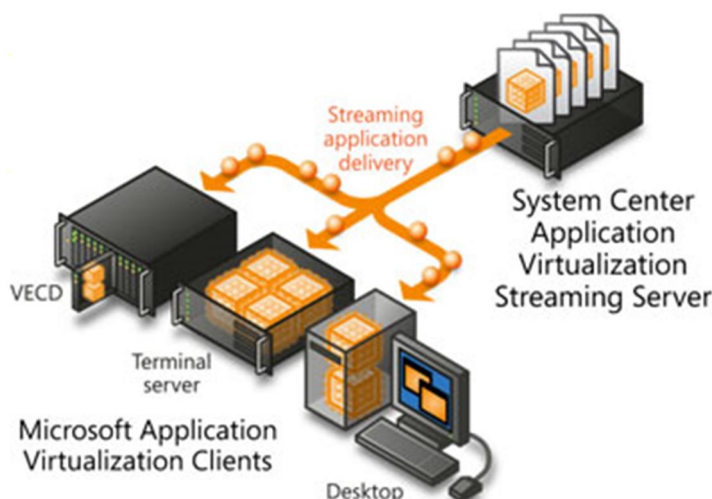
Itsenäisen mallin käyttämistä kannattaa pohtia varsinkin silloin, kun organisaatiossa on yhteydettömiä tai hitaan yhteyden päässä olevia etäkäyttäjiä. Itsenäinen malli sopii myös hyvin yhteen olemassa olevien sovellusten hallintajärjestelmien kanssa, koska virtuaalisovellus on paketoitu MSI-tiedostoksi. (Microsoft 2010a, 11-12.)

Edellä mainittujen tapauksien lisäksi itsenäistä mallia kannattaa hyödyntää virtualisoidun sovellusten testauksessa. Testausta varten ei tarvitse rakentaa monimutkaista ympäristöä: tarvitaan vain kaksi virtuaalista työasemaa. Toinen työasemista hoitaa sovelluksen virtualisoinnin, kun taas toista voidaan käyttää testaamiseen. (Mangan 2009, 3.)

Suoratoistomalli

App-V käyttää suoratoistomallissa yhtä tai useampaa suoratoistopalvelinta, sekvensseriä ja asiakasohjelmaa. Kuvio 6 havainnollistaa suoratoistomallin vaatimaa ympäristöä. Luotu MSI-paketti pitää jakaa erikseen käyttäjille esimerkiksi sovellusten hallintajärjestelmällä. MSI-paketti sisältää kaikki muut virtualisointitiedostot lukuun otta-

matta SFT-tiedostoa, joka tuodaan käyttäjän saataville suoratoiston avulla. Sovelluksesta suoratoistetaan ensiksi lohko FB1, jotta sovellus saadaan nopeammin käyttäjän saataville. (Buller ym. 2010, 306.)



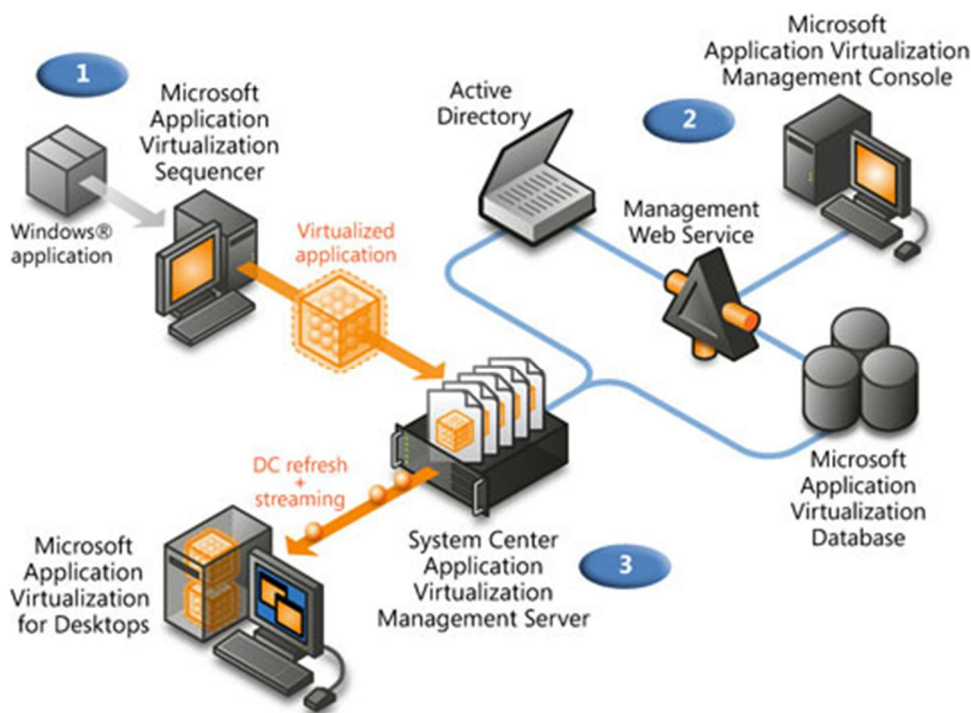
KUVIO 6. Suoratoistomalli (Microsoft 2011)

Suoratoistomallin suurimmat hyödyt ja samalla rajoitukset ovat hallintapalvelimen sekä SQL-lisenssien pois jääminen. Uusia palvelimia ei tarvitse välttämättä tehdä ollenkaan, sillä suoratoistomalli yhdistäminen on helppoa käytössä oleviin teknologioihin ja toimintatapoihin. Ilman hallintapalvelinta App-V ei pysty hallitsemaan sovellusten käyttöä tai päivityksiä, jolloin joudutaan käyttämään esimerkiksi sovellusten hallintajärjestelmää. Lisäksi ei voida hyödyntää hallintapalvelimen tarjoamaa lisenssien valvontaa eikä raportointi- ja mittarointityökaluja. (Buller ym. 2010, 306.)

Täyden infrastruktuurin malli

App-V:n täyden infrastruktuurin malli koostuu hallintapalvelimesta, joka sisältää myös SQL-tietokannan, sekvensseristä, suoratoistopalvelimesta ja asiakasohjelmasta. Kuvio 7 havainnollistaa täyden infrastruktuurin mallin toimintaa. Suoratoistopalvelinta ei ole pakko käyttää, sillä hallintapalvelin voi toimia myös suoratoistopalvelimena. Hallintapalvelinta käytetään sovellusten julkaisemiseksi asiakasohjelmiin. Julkaisu-

vaiheessa määritetään sovelluksen käyttöoikeudet aktiivihakemistossa olevien ryhmien avulla. Kun käyttöoikeuden omaava käyttäjä kirjautuu sisään asiakasohjelman sisältämälle työasemalle, käyttäjälle määritettyjen sovellusten kuvakkeet ilmestyvät työpöydälle tai aloitusvalikkoon määritettyjen asetusten mukaan. (Alvarez 2011, 32.)



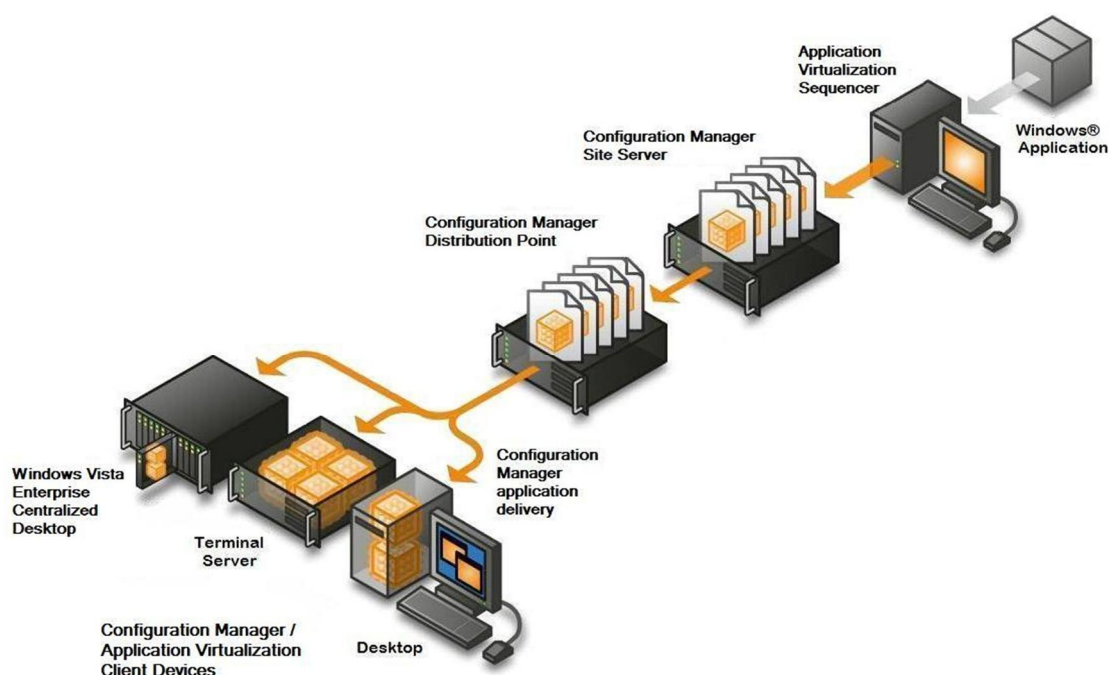
KUVIO 7. Täyden infrastruktuurin malli (Microsoft 2011)

Täyden infrastruktuurin malli tarjoaa edellä mainittuihin ratkaisuihin nähden kaksi tärkeää ominaisuutta: raportointi ja keskitetty lisenssien hallinta. Raportointi mahdollistaa jokaisen sovelluksen seurannan. Raporteista nähtävän käyttöasteen perusteella voidaan tarpeen vaatiessa tehdä muutoksia järjestelmän parantamiseksi. Keskitetyn lisenssien hallinnan avulla sovelluksia voi käyttää vain niihin oikeutetut käyttäjät ja käytettyjen lisenssien määrä on aina tiedossa. (Mts. 32.)

SCCM & App-V

Integroimalla App-V SCCM-järjestelmään saavutetaan kokonaisvaltainen jakelu- ja päivityspalvelu virtualisoiduille sovelluksille. SCCM:ää käytettäessä App-V-

infrastruktuuri koostuu tyypillisesti pelkästään sekvensseristä ja asiakasohjelmasta. Kuviosta 8 nähdään, kuinka SCCM ja App-V toimivat yhdessä. Sovellukset virtualisoidaan tavalliseen tapaan sekvensserillä, jonka jälkeen ne siirretään Configuration Manager Site Serverille ja sieltä edelleen jakelupisteisiin. Jakelupisteiltä sovellukset tuodaan käyttäjän saataville suoratoistamalla tai paikallisella jakelulla. Sovellukset voidaan kohdistaa jaeltavaksi käyttäjä- tai konekohtaisesti. (Microsoft 2010c, 5-6.)



KUVIO 8. SCCM & App-V yhdessä (Microsoft 2010c, 6)

App-V:n integroinnilla SCCM-järjestelmään saavutetaan monia hyötyjä verrattuna App-V:n omiin toimintamalleihin. SCCM:ää käytettäessä sekä perinteisiä että virtualisoituja sovelluksia voidaan hallita keskitetysti samaa järjestelmää käyttäen. SCCM skaalautuu suurien yritysten tarpeiden mukaan. Yksi suurimmista eduista on sovellusten jakelu myös konekohtaisesti. SCCM:n huonoja puolia ovat esimerkiksi pikakuvakkeiden ja tiedostotyyppien välisten assosiaatioiden hallinta. Pikakuvakkeiden sijaintia tai tiedostotyyppien välisiä assosiaatioita ei voida muokata SCCM:llä vaan kyseiset toimet joudutaan suorittamaan sekvensserillä. Huomionarvoinen seikka on

myös raportointiominaisuuksien rajoittuneisuus täyden infrastruktuurin malliin verrattuna. (Mts. 10–11.)

4.1.5 Rajoitukset

App-V:llä ei yleensä pystytä virtualisoimaan seuraavia sovellustyyppejä (Alvarez 2011, 120-121):

- Sovellukset, jotka sisältävät laiteajureita
- Sovellukset, jotka ovat sidottuja käyttöjärjestelmään (Internet Explorer)
- Sovellukset, jotka käyttävät komentotulkin laajennuksia (WinZip)
- COM+-sovellukset (BizTalk)
- Sovellukset, jotka suorittavat tausta-ajaja (Palomuuuri)
- Sovellukset, jotka ovat riippuvaisia monista muista sovelluksista (Microsoft Office Communicator 2007)
- Sovellukset, joiden lisensointi on sidottu laitteistoon (Computer-aided design)
- Sovellukset, joiden SFT-tiedosto vie yli neljä gigatavua levytilaa
- 16-bittiset sovellukset 64-bittisessä käyttöjärjestelmässä.

4.1.6 Sovellusten linkitys

Virtualisoidut sovellukset pystyvät hyödyntämään käyttöjärjestelmään perinteisesti asennettuja sovelluksia, mutta ne eivät näe toisiaan automaattisesti. Ongelma voidaan ratkaista esimerkiksi paketoimalla sovellukset samalla nauhoituskerralla. Kyseinen ratkaisu ei ole kuitenkaan viisas ratkaisu pitkällä tähtäimellä, sillä pakettien koot voivat kasvaa suuriksi tai samoja sovelluksia voidaan joutua virtualisoimaan useita kertoja. (Alvarez 2011, 187; Buller ym. 2010, 283.)

Parempi ratkaisu ongelmaan on virtualisoitujen sovellusten linkitys DSC:n (Dynamic Suite Composition) avulla. DSC:tä käyttämällä virtualisoitu sovellus voidaan linkittää sitä tarvitseviin toisiin virtuaalisovelluksiin, jolloin sovellusta ei tarvitse virtualisoida kuin yhden kerran. Linkkien muodostaminen on yksinkertaista OSD-tiedoston muok-

kaamista. Muokkaaminen voidaan tehdä käsin tai käyttämällä Dynamic Suite Composition tool-työkalua. (Alvarez 2011, 188.)

Dynamic Suite Composition jakaa sovellukset kahteen eri kategoriaan: pää- ja toissijaisiin sovelluksiin. Pääsovelluksella tarkoitetaan sovellusta, jota käytetään ennen toissijaisia sovellusta. Toissijainen sovellus on usein pääsovellukseen liittyvä lisäosa (plugin), jota pääsovellus käyttää tarvittaessa. DSC:tä käytettäessä on myös huomiotava, että pääsovellus voi olla riippuvainen lukuisista toissijaisista sovelluksista, mutta toissijaiset sovellukset eivät voi riippuvaisia toisistaan. (Mts. 188.)

4.2 VMware ThinApp

4.2.1 Yleistä

ThinApp (aiemmin Thinstall) on VMwaren ratkaisu sovellusvirtualisointiin. VMware hankki Thinstallin itselleen yritystollalla tammikuussa 2008 ja pääsi siten osaksi sovellusvirtualisointimarkkinoita. (Madden 2008.) VMware on kehittänyt ThinAppia siitä lähtien ja tällä hetkellä ollaan versiossa 4.6. Versio 4.6 toi mukanaan esimerkiksi tuen Internet Explorer 6:n käyttämiseen Windows 7 -käyttöjärjestelmässä. (VMware 2010c.)

ThinApp on agentiton ratkaisu, joten se ei vaadi erillisen asiakasohjelman asennusta työasemaan. Se on suunniteltu integroitavaksi jo olemassa olevaan infrastruktuuriin, joten erillisiä servereitä tai tietokantoja ei tarvitse tehdä. ThinApp paketoivat virtualisoidun sovelluksen asetuksineen yhdeksi ainoaksi suoritettavaksi tiedostoksi, jota voidaan käyttää lukuisissa eri Windows-käyttöjärjestelmissä. ThinApp:lla paketoitujen sovellukset toimivat käyttäjätason oikeuksilla, joten sovellusta voidaan käyttää kaikissa koneissa turvallisesti. (VMware 2010a, 3-4.)

4.2.2 Komponentit

ThinApp:n käyttö tapahtuu käytännössä yhden komponentin avulla, sillä se ei sisällä App-V:n tapaan asiakasohjelmaa tai hallinta- ja suoratoistopalvelimia. ThinApp käyttää Setup Capture-sovellusta sovelluksen asennuksen nauhoittamiseen ja sovelluksen virtualisointiin. Muita oleellisia ThinApp:n asennuksessa tulevia sovelluksia ovat (VMware 2010b, 13-14):

- log_monitor.exe, joka näyttää sovelluksen suoritushistorian sekä virheet
- relink.exe, jolla olemassa olevat paketit päivitetään uusimpaan ThinApp:n versioon
- snapshot.exe, joka vertaa ympäristöä ennen sovelluksen asennusta sekä sen jälkeen
- thinreg.exe, joka rekisteröi nauhoitetut sovellukset työasemaan. Rekisteröinti sisältää pika- ja käynnistysvalikon kuvakkeiden luonnin sekä tiedostotyyppien väliset assosiaatiot
- tlink.exe, joka yhdistää nauhoitetun sovelluksen olennaisimmat moduulit paketin luontivaiheessa
- vftool.exe, joka kokoaa virtuaalitiedostojärjestelmän paketin luontivaiheessa
- vregtool.exe, joka kokoaa virtuaalirekisterin paketin luontivaiheessa.

4.2.3 Sovelluksen virtualisoinnin vaiheet

Sovelluksen virtualisointi on monivaiheinen prosessi. Setup Capture tuottaa virtualisointiprosessin aikana lukuisia eri tiedostoja. Prosessin aikana esille tulee myös monenlaisia termejä, kuten projekti, package.ini, luontiprosessi, build.bat, paketti, eristystila, hiekkalaatikko ja aloituspiste. (VMware 2009, 8-10.)

Projekti

Projekti (project) tarkoittaa Setup Capture-prosessin tuottamaa tietoa. Projekti on tärkeä osa virtualisoitujen sovellusten hallinnassa, koska se sisältää kaikki sovellusten luomiseen ja muokkaamiseen tarvittavat tiedot. (Mts. 8-9.)

Package.ini

Package.ini-tiedosto löytyy jokaisesta projektista. Se sisältää Setup Capturen nauhoittamat asetukset sekä muut Setup Capturen ulkopuolella tehdyt konfiguraatiot. Kaikkea package.ini:n sisältää tietoa voidaan muokata yksinkertaisesti tekstieditorin avulla.

(Mts. 9.)

Luontiprosessi

Luontiprosessissa (build) projekti-hakemistot sekä konfiguraatiot tiivistetään ja sisällytetään pakettiin. Se voidaan suorittaa Setup Capture-prosessin viimeisenä vaiheena tai vaihtoehtoisesti käyttämällä projekti-hakemistosta löytyvää build.bat-tiedostoa.

Luontiprosessi voidaan suorittaa milloin tahansa uudestaan uusien asetusten sisällyttämiseksi pakettiin. Paketti voidaan luoda uudestaan millä tahansa työasemalla, jolla on pääsy projekti-hakemistoon sekä ThinApp:n vftool.exe-, vfregtool.exe- ja tlink.exe-tiedostoihin. (Mts. 9.)

Build.bat

Build.bat on eräajotiedosto, jota käytetään paketin luomiseen joko manuaalisesti tai Setup Capture-prosessin aikana. (Mts. 9.)

Paketti

Paketti (package) on nauhoitus- ja luontiprosessin tulos. Se sisältää kaikki virtualisoidun sovelluksen suorittamiseen tarvittavat komponentit. Luotu exe-päätteinen paketti on pakattu sekä suojattu salauksella mahdollisten muokkausyritysten varalta. Paketti voidaan sisällyttää myös MSI-tiedostoon, jolloin sen jakelu voidaan suorittaa ESD-järjestelmällä (Electronic Software Delivery). MSI-tiedosto sisältää itse paketin, thin-reg.exe:n sekä msi-tietokannan. (Mts. 9.)

Eristystila

Eristystilalla (Isolation Mode) määritetään virtuaalisovelluksen ja käyttöjärjestelmän välisen eristyksen määrä. Asetuksia muuttamalla säädetään, mitkä käyttöjärjestelmän resurssit ovat näkyviä virtuaalisovellukselle. Eristystila voi olla yhdistetty (merged), kirjoita-kopioi (write-copy) sekä täysi (full). (Mts. 10.)

Hiekkalaatikko

Hiekkalaatikko (sandbox) on käyttäjäkohtainen kansio. Se luodaan niiden virtuaalisten sovellusten varalle, jotka tekevät muutoksia virtuaalirekisteriin, -kansioihin ja -tiedostoihin. Virtualisoitu sovellus ylläpitää hiekkalaatikkoon ja käyttää apunaan eristystilan asetuksia määrittääkseen, kirjoitetaanko tieto hiekkalaatikkoon vai paikalliseen käyttöjärjestelmään. Hiekkalaatikko sijaitsee oletuksena käyttäjän %AppData% -hakemistossa, mutta se on muutettavissa. (Mts. 10.)

Aloituspiste

Aloituspisteellä (Entry Point) tarkoitetaan exe-tiedostoa, joka käynnistyy pikakuvakkeesta. Aloituspisteiden määrä riippuu käyttäjän tarpeista ja nauhoitetun sovelluksen ominaisuuksista. (VMware 2010b, 17.) Esimerkiksi käytettäessä Microsoft Officea voidaan luoda aloituspisteet Excelille ja Wordille. (Mts. 10.)

Sovelluksen virtualisointi

Sovelluksen virtualisointi aloitetaan nauhoitusprosessilla ja päätetään luontiprosessiin. Koko prosessi suoritetaan samalla Setup Capture-ohjelmalla. Nauhoitusprosessin aikana luodaan projekti sovelluksen ja konfiguraatioiden tallentamiseksi. Luontiprosessi tiivistää ja sisällyttää projekti-hakemistot ja konfiguraatiot pakettiin. Luotuja paketteja voidaan muokata tai päivittää projekti-hakemistosta löytyvien tiedostojen avulla. (VMware 2010a, 6.)

Ennen nauhoitusprosessia on hyvä selvittää sovelluksen tarpeet ja riippuvuudet. Sovellus voi olla riippuvainen esimerkiksi muista sovelluksista tai kirjastoista. Riippuvuudet voidaan nauhoittaa samalla nauhoituskerralla tai käyttämällä Application Link-tekniikkaa linkkien muodostamiseen riippuvaisten tekijöiden välillä. (VMware 2010b, 15.)

Nauhoitusprosessi aloitetaan käyttöjärjestelmän skannaamisella vertailukohdan muodostamiseksi. Seuraavaksi asennetaan virtualisoitava sovellus ja muokataan sitä tarpeiden mukaan. Lopuksi skannaus tehdään uudestaan ja saatua tulosta verrataan alkuperäiseen. Saatujen eroavaisuuksien perusteella muodostetaan virtuaalisovellus. (Mts. 16-17.)

Nauhoitusprosessin jälkeen määritetään joukko virtualisoituun sovellukseen liittyviä asetuksia. Ensimmäisenä valitaan aloituspisteet sovelluksen käynnistymistä varten. Seuraavaksi määritetään käyttöoikeudet paketin suorittamista varten. Käyttöoikeus voidaan antaa kaikille käyttäjille tai valituille ryhmille hyödyntäen aktiivihakemistossa määriteltyjä ryhmiä. (VMware 2009, 13-14.) Muita määritettäviä asetuksia ovat eristystila, hiekkalaatikon sijainti, projektin sijainti ja nimi sekä tiedon säilytyspaikka (data container). Lisäksi voidaan valita, luodaanko erillinen MSI-paketti ja käytetäänkö pakkausta tiedoston koon pienentämiseksi. (Mts. 15-17.)

Lopuksi suoritetaan vielä luontiprosessi, jonka jälkeen virtualisoitu sovellus on käytettävissä. Ennen luontiprosessia on vielä mahdollista muokata Package.ini-tiedostoa esimerkiksi linkkien määrittämiseksi sovellusten välille. (Mts. 17-18.)

4.2.4 ThinAppin toimintamallit

ThinApp tarjoaa kaksi eri toimintamallia sovellusten käyttämiseen: Streamed Execution Mode ja Deployed Execution Mode. Malleja voi käyttää myös samaan aikaan, jolloin voidaan hyödyntää molempien tarjoamat edut.

Streamed Execution Mode

Suoratoistomallissa käytettävä sovellus sijaitsee jaetussa hakemistossa, josta se tuodaan sitä tarvitsevien käyttäjien saataville suoratoistamalla. Suoratoistomalli on paras vaihtoehto keskitetyissä ympäristöissä, joissa työasemat ovat aina kytkettyinä verkkoon. Loppukäyttäjä käynnistää sovelluksen suoratoiston esimerkiksi työpöydältä tai käynnistä-valikosta löytyvällä pikakuvakkeella, joka osoittaa virtualisoituun sovellukseen jaetussa hakemistossa. (VMware 2009, 23.)

Suoratoistoa käytettäessä sovellusten redundanttisuuteen pitää kiinnittää erityistä huomiota, jotta sovellukset olisivat aina saatavilla. Saatavuus voidaan turvata esimerkiksi Microsoftin DFS-tekniikalla (Distributed File System). Lisäksi verkkoliikenteen kasvun määrän vaikutukset tulee huomioida. (Mts. 23.)

Suoratoistomallin suurin etu on keskitetty hallinta. Virtualisoitu sovellus saadaan käyttäjän saataville helposti esimerkiksi sisäänkirjautumisen aikana suoritettavien komentosarjojen (login script) avulla. Komentosarjalla voidaan käynnistää thin-reg.exe, joka lisää käyttäjälle tarvittavat pikakuvakkeet virtuaalisovelluksen käynnistämistä varten. Suoratoistomalli sopii käyttäjille, jotka käyttävät useita eri työasemia. Käyttökokemus pysyy samana mitä tahansa työasemaa käytettäessä, koska samat ohjelmat ovat aina saatavilla. (Mts. 23.)

Deployed Execution Mode

Deployed Execution Mode:ssa paketit pitää jakaa loppukäyttäjälle ennen käyttöä. Tästä johtuen sovellus on aina käytettävissä. Paketti voi sijaita esimerkiksi käyttäjän työaseman kiintolevyllä tai USB-muistitikulla. Paketin jakaminen käyttäjälle voidaan hoitaa helposti edellä mainittujen tallennusmedioiden avulla tai sovellusten hallintajärjestelmillä, joita useimmat organisaatiot käyttävät. Deployed Execution Moden hyödyt ovat sovelluksen toiminnan ja saatavuuden riippumattomuus verkosta tai tiedostopalvelimista. (VMware 2009, 23-24.)

4.2.5 Rajoitukset

ThinApp ei pysty virtualisoimaan kaikkia sovelluksia. Esimerkiksi seuraavien sovellystyyppien virtualisointi ei onnistu tai ne eivät toimi oikein (VMware 2010b, 12):

- Sovellukset, jotka vaativat laiteajureita toimiakseen
- Virustorjuntaohjelmat ja palomuurit
- Osa VPN-asiakasohjelmista
- DCOM:ia verkon yli käyttävät sovellukset
- Sovellukset, jotka tarjoavat komentotulkki-integraation.

4.2.6 Sovellusten linkitys

ThinApp:n sisältämä Application Link mahdollistaa linkkien muodostamisen virtualisoitujen sovellusten välille, jolloin toisistaan riippuvaisia sovelluksia ei tarvitse pake-

toida samalla nauhoituskerralla. Tästä modulaarisesta ratkaisusta johtuen pakettien koot pysyvät pienempinä. Lisäksi olemassa olevien sovellusten päivitys helpottuu, sillä sovelluksen päivittämiseen riittää ainoastaan yhden sovelluksen nauhoittaminen uudelleen. Jos sovellukset olisi nauhoitettu samaan pakettiin, ne kaikki jouduttaisiin asentamaan uudestaan. Tyypillisesti Application Link:iä käytetään sovelluksissa, joista monet muut sovellukset ovat riippuvaisia. Näitä sovelluksia ovat esimerkiksi runtime-komponentit, kuten .NET ja JRE (Java Runtime Environment). (VMware 2009, 30.)

Luotujen sovellusten linkitys on helppoa. Linkitys tehdään muokkaamalla toisesta sovelluksesta riippuvaisen sovelluksen package.ini-tiedoston *RequiredAppLinks*-arvoa. Jos arvo on esimerkiksi *RequiredAppLinks*=FoxitReader.exe, luo sovellus linkin FoxitReader-sovellukseen. Sovellusten täytyy sijaita samassa hakemistossa, jotta linkitys toimii. Linkin tila voi olla vaadittu (required) tai valinnainen (optional). Linkin ollessa vaaditussa tilassa sovellus ei käynnisty, jos se ei voi yhdistää linkitettyyn sovellukseen. (Mts. 30.)

5 TESTAUS

5.1 Testiympäristö

Virtualisointiratkaisujen testausta varten luotiin testiympäristö, joka koostui yhdestä Windows 7 -käyttöjärjestelmän sisältävästä kannettavasta tietokoneesta ja seuraavista virtuaalikoneista:

- Yksi Windows Server 2008-palvelin, jossa oli yksi gigatavu muistia.
- Kaksi Windows XP –työasemaa, joissa oli yksi gigatavu muistia.
- Kaksi Windows 7 –työasemaa, joissa oli yksi gigatavu muistia.

Sekä kannettavan tietokoneen että virtuaalikoneiden käyttöjärjestelmät olivat 32-bittisiä.

Windows Server 2008-Palvelin ja Windows 7-työasemat liitettiin toimeksiantajan toimialueeseen (domain). Kaikista virtuaalikoneista otettiin palomuri pois käytöstä testauksen nopeuttamiseksi. Olemassa olevaan aktiivihakemistoon luotiin appv-kayttaja niminen käyttäjä sekä ryhmät AppV_Admins ja AppV_Users molemmilla tekniikoilla virtualisoitujen sovellusten käyttöoikeuksien testaamiseksi. Appv-kayttaja lisättiin AppV_Users-ryhmään, johon sovellusten käyttöoikeus kohdistettiin.

XP-työasemia käytettiin sovellusten virtualisointiin. XP1-työasemaan lisättiin toinen kiintolevy, jolle annettiin levytunnus Q App-V:n suositusten mukaan. XP1-työasemaan asennettiin sekvensseri. ThinApp ei vaatinut erityistoimenpiteitä, joten sen käyttämään XP2-työaseman kokoonpanoon ei tarvinnut tehdä muutoksia. XP2-työasemalle asennettiin ThinApp Setup Capture-ohjelman käyttämiseksi. XP-työasemille ei asennettu ylimääräisiä ohjelmia tai päivityksiä, jotta työaseman käyttöjärjestelmä pysyisi mahdollisimman puhtaana. Näin varmistettiin, että sovellukset asentavat varmasti kaikki tarvittavat komponentit asennuksen aikana. Tarpeellisten asennusten jälkeen XP-työasemista otettiin tilannekuvat (snapshot), joiden avulla käyttöjärjestelmä voitiin palauttaa pisteeseen ennen sovellusten asennusta. Tilannekuvia käyttämällä virtualisoitujen sovellusten testausta voitiin nopeuttaa merkittävästi.

Windows 7 -työasemia käytettiin virtualisoitujen sovellusten toimivuuden sekä käyttöoikeuksien testaamiseen molemmilla tekniikoilla. Windows 2008 -palvelimelle asennettiin vain App-V-hallintapalvelin ja sen vaatimat komponentit, sillä ThinApp ei vaatinut erillisiä asennuksia palvelimelle.

5.2 Virtualisoitavat sovellukset ja niiden testaus

Testattavia sovelluksia olivat Firefox 4.0, vSphere Client, Foxit Reader, Uranus (HUS:ssa käytössä oleva potilastietojärjestelmä) sekä Microsoft Office 2010, joista Microsoft Office ja Uranus sisältävät useita eri sovelluksia. Jokaista sovellusta testattiin verkon yli suoratoistamalla ja paikallista asennusta käyttämällä. Sovellukset virtualisoitiin tarpeen mukaan useita kertoja erilaisilla asetuksilla toimivan kokonaisuuden löytämiseksi. Jokainen sovellus virtualisointiin vähintään kaksi kertaa pakkauksen vaikutuksen määrittämiseksi. Uranus-sovelluksen asennus ja testaus on nähtävissä

esimerkkinä kummankin sovellusvirtualisointitekniikan yhteydessä. Sovelluksia testattiin käyttämällä niitä samaan tapaan kuin perinteisesti asennettuja. Kaikkien sovellusten sisältämien ominaisuuksien toimintaa ei lähdetty tutkimaan, vaan keskityttiin käyttämään yleisimpiä ominaisuuksia.

5.3 App-V

App-V:tä testattiin itsenäisellä ja täyden infrastruktuurin mallilla. Näin saatiin tietoa sekä paikallisesta että suoratoistetusta toiminnasta. Täyden infrastruktuurin mallin testauksessa ei käytetty erillistä suoratoistopalvelinta, sillä hallintapalvelin riitti täysin testauksen suorittamiseen. Testauksessa kiinnitettiin huomiota mallien toimintakuntoon saattamisen vaativuuteen, käytettävyyteen ja virtualisoitujen sovellusten toimintaan.

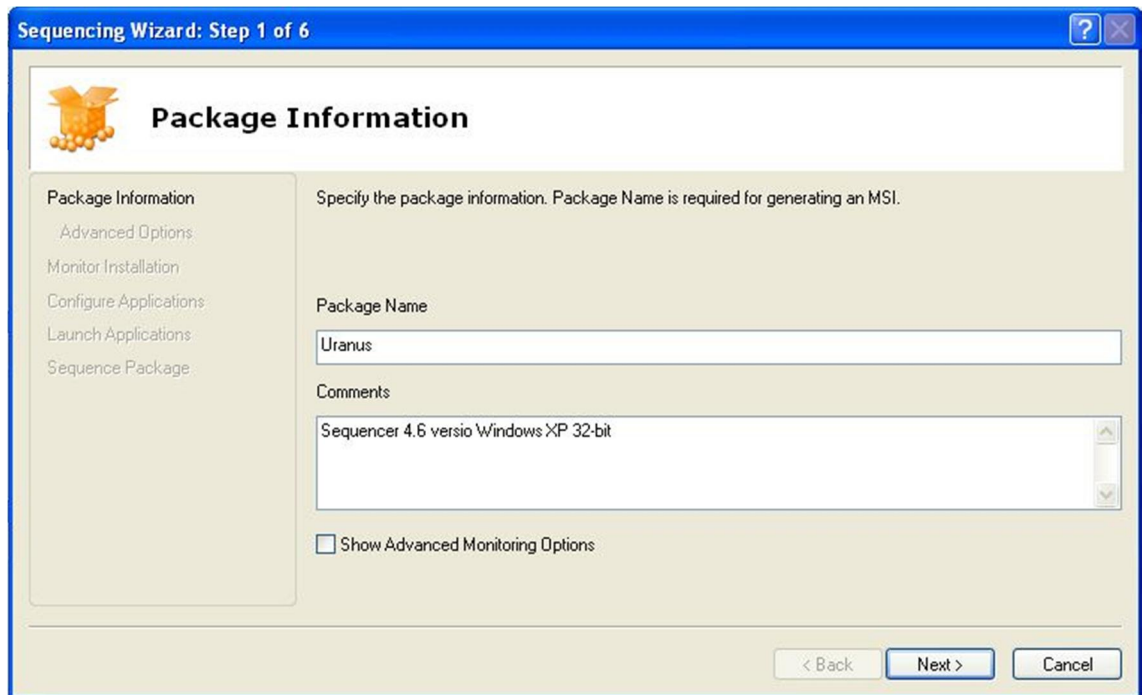
5.3.1 Sekvensserin käyttö

Sovelluksen virtualisointi alkoi uuden paketin luonnilla kuvion 9 mukaisesti. Kuviosta nähdään myös muut sekvensserin antamat vaihtoehdot, joita ovat olemassa olevan paketin muokkaaminen tai päivittäminen.



KUVIO 9. Paketin luonti, muokkaus tai päivitys

Kuviosta 10 nähdään paketin luonnin ensimmäinen vaihe, jossa annettiin nimi ja kommentteja. Kommentteihin kirjoitettiin tässä tapauksessa sekvensserin versionumero sekä käyttöjärjestelmä, jossa sekvensseriä käytetään. Show Advanced Monitoring Options tarjoaa lisäasetuksia nauhoitukseen. Sitä ei kuitenkaan käytetty, koska suurimmalle osalle ohjelmista suositellaan oletusasetuksia.



The image shows a screenshot of the 'Sequencing Wizard: Step 1 of 6' window. The title bar is blue with the text 'Sequencing Wizard: Step 1 of 6' and a help icon. The main window has a light beige background. On the left, there is a vertical list of steps: 'Package Information' (selected), 'Advanced Options', 'Monitor Installation', 'Configure Applications', 'Launch Applications', and 'Sequence Package'. The main area is titled 'Package Information' and contains the instruction: 'Specify the package information. Package Name is required for generating an MSI.' Below this, there is a 'Package Name' text box containing the text 'Uranus'. Underneath that is a 'Comments' text box containing the text 'Sequencer 4.6 versio Windows XP 32-bit'. At the bottom of the main area, there is a checkbox labeled 'Show Advanced Monitoring Options' which is currently unchecked. At the bottom right of the window, there are three buttons: '< Back', 'Next >', and 'Cancel'.

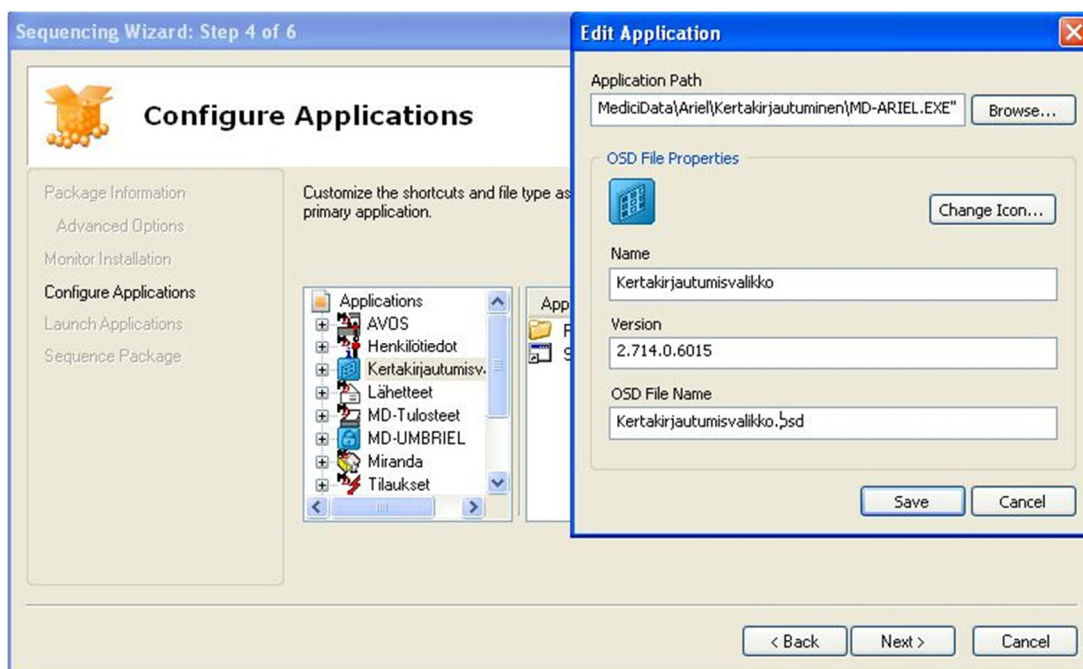
KUVIO 10. Paketin nimeäminen

Kuviosta 11 nähdään valittu monitoroitava kansio, joka oli tässä tapauksessa uranus.v01. Monitoroitavan kansion pitää olla suoraan aseman juuressa. Microsoft suosittelee käytettäväksi 8.3-nimeämistapaa kansion nimeämiseen. Sovelluksen nimi saa siis sisältää enintään kahdeksan merkkiä ja mahdollisen pisteen jälkeen saa olla enintään kolme merkkiä. Esimerkiksi "Q:\Uranus" tai "Q:\Uranus.v01" on oikein, kun taas "Q:\Uranus Sovellus" on väärin.



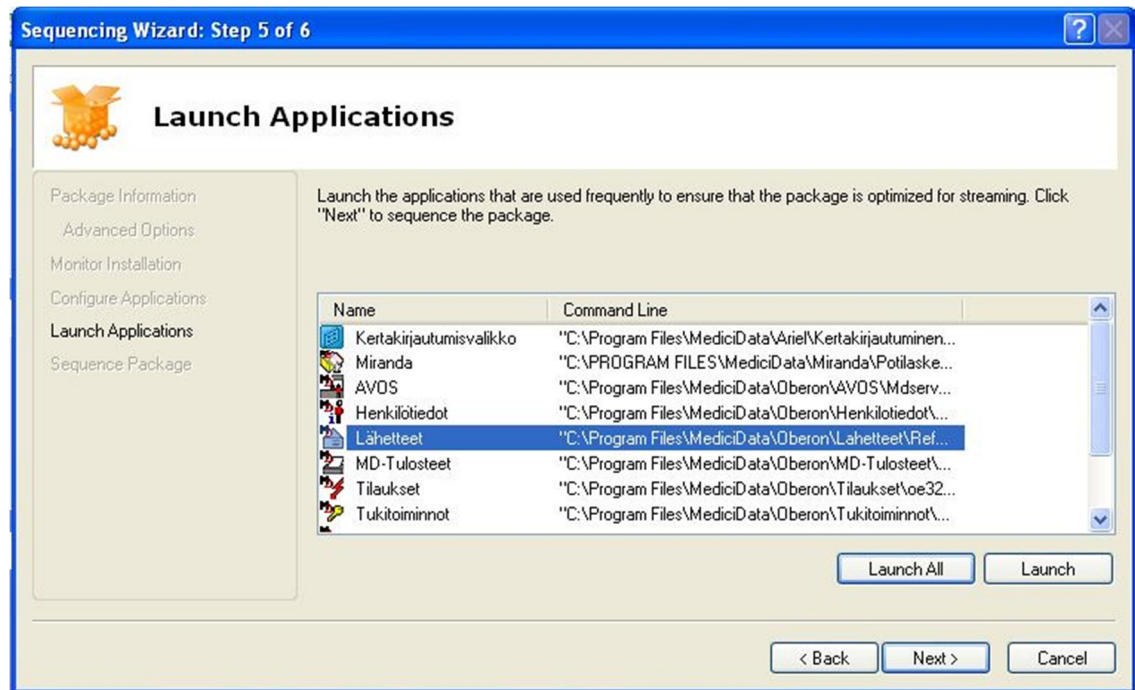
KUVIO 11. Monitorointi

Kuviosta 12 nähdään monitoroinnin nauhoittamat sovellukset sekä esimerkki sovelluksen muokkaamisesta. Sovelluksia voi myös lisätä tai poistaa samasta näkymästä. Jokaisen sovelluksen OSD-tiedoston nimestä poistettiin välit, koska ne aiheuttavat joskus ongelmia sovellusten julkaisussa. Sovellusta pääsee muokkaamaan valitsemalla sovelluksen hiiren oikealla napilla.



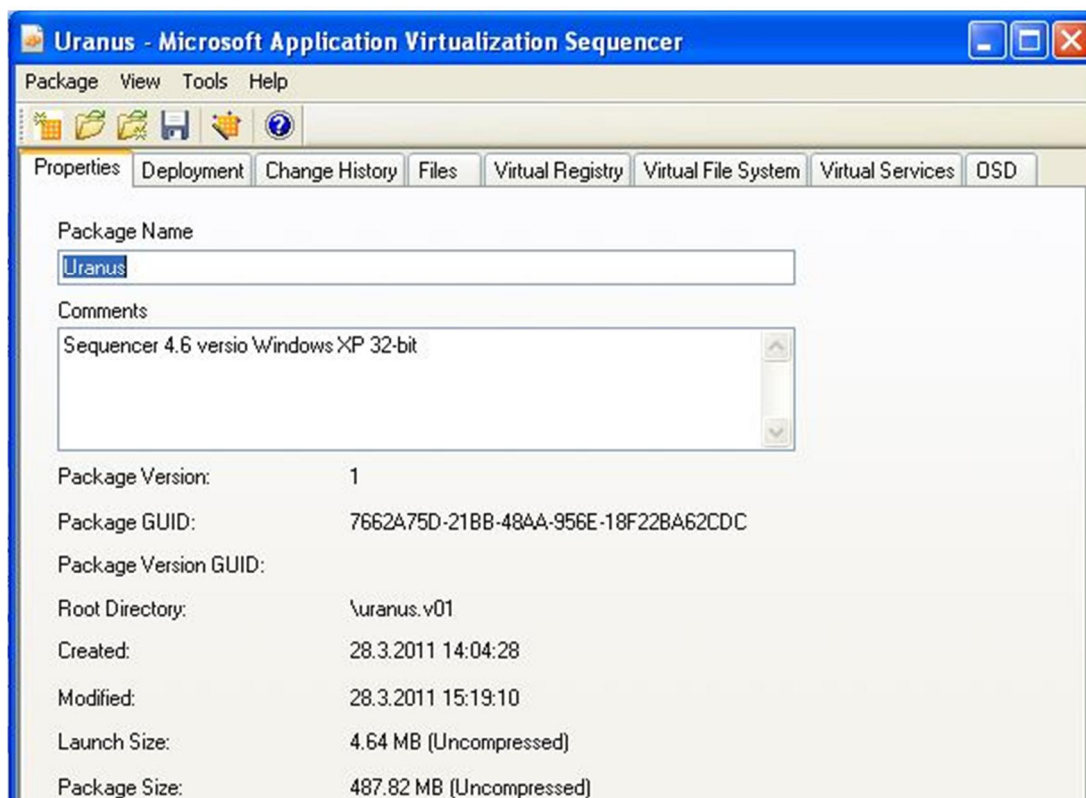
KUVIO 12. Monitoroinnissa havaitut sovellukset

Seuraavassa vaiheessa käynnistettiin halutut sovellukset FB1-lohkon muodostamiseksi, kuten kuviosta 13 voi havaita. Samalla pystyttiin todentamaan, oliko sovellus asennut oikein käynnistettävien sovellusten kohdalla. Lopuksi sekvensseri kokosi paketin yhdeksi kokonaisuudeksi.



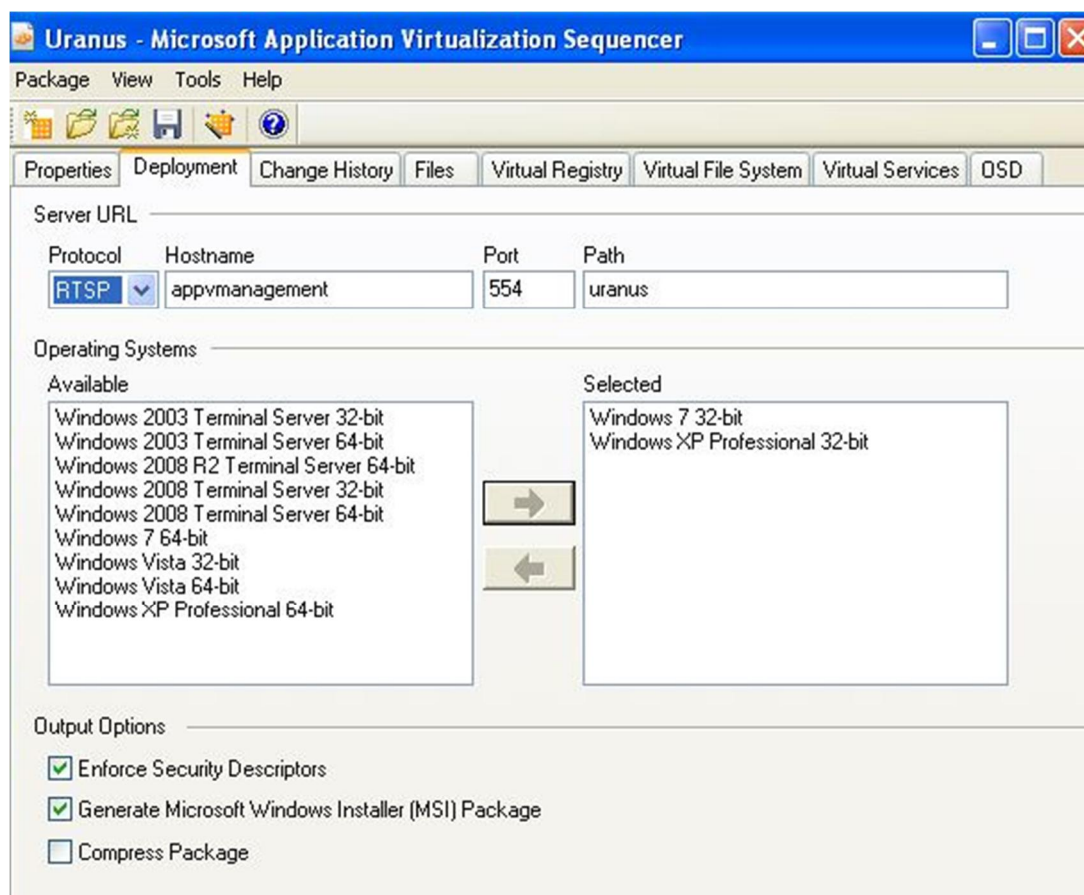
KUVIO 13. FB1-lohkon muodostus

Paketoinnin jälkeen siirryttiin viimeiseen vaiheeseen, jossa paketin sisältämää tietoa voitiin käydä läpi välilehdittäin. Kuvio 14 nähdään luodun paketin yleiset tiedot, kuten paketin ja käynnistyslohkon koko sekä muut tietoa sisältävät välilehdet.



KUVIO 14. Paketin tiedot

Deployment-välilehdellä määritettiin jakeluun liittyvät asetukset, jotka voi havaita kuvista 15. Käytettäväksi protokollaksi valittiin RTSP, jolloin käytettävän portin arvoksi tulee automaattisesti 554. Suoratoistopalvelimen nimeksi määritettiin appv-management ja path-kohtaan kirjoitettiin hakemisto, jossa luotava paketti tulee sijaitsemaan content-hakemiston sisällä. Seuraavaksi määritettiin käytettävät käyttöjärjestelmät, jotka olivat tässä tapauksessa Windows 7 32-bit sekä Windows XP Professional 32-bit. Ennen paketin tallennusta aktivoitiin vielä kohdat Enforce Security Descriptors ja Generate Microsoft Installer (MSI) Package. Ensimmäisenä mainittua valintaa käyttämällä asennetun sovelluksen NTFS-oikeudet kopioidaan myös luotavaan pakettiin. Jälkimmäisen avulla luodaan erillinen MSI-tiedosto, jota käytettiin itsenäisen mallin testaamiseen. Paketti tallennettiin työpöydälle uranus2-nimiseen hakemistoon odottamaan siirtoa hallintapalvelimelle.



KUVIO 15. Sovelluksen jakeluun liittyviä asetuksia

5.3.2 Täyden infrastruktuurin malli

Alkutoimenpiteet

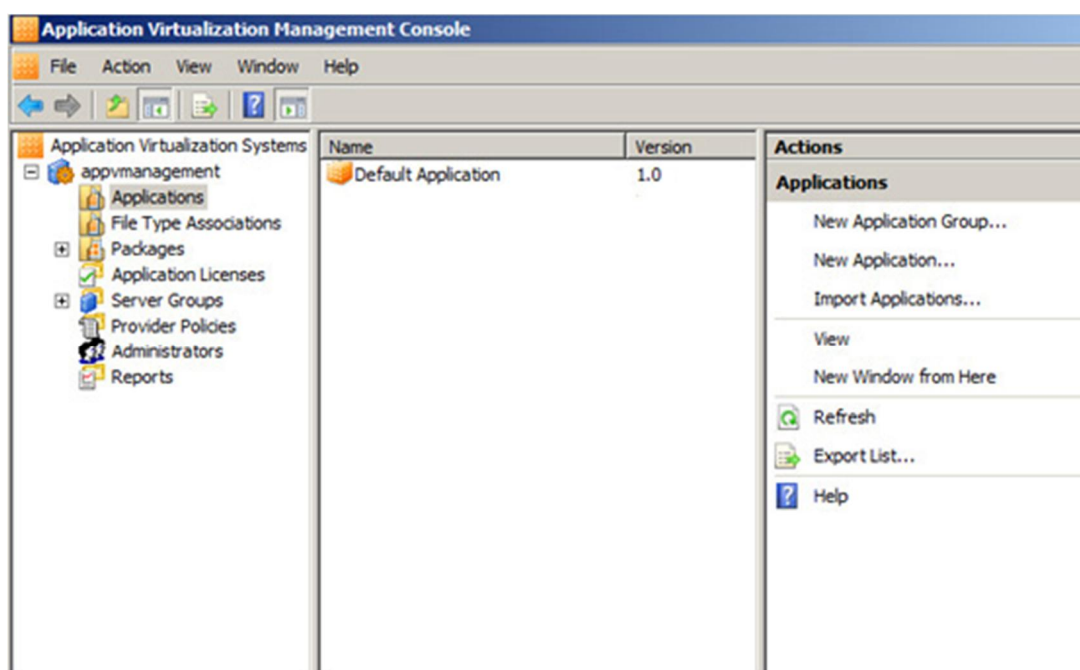
Windows-palvelimelle asennettiin Internet Information Services-rooli sekä SQL Server 2008 R2 Express Edition ennen hallintapalvelimen asennusta. Kyseisten komponenttien asennus on kuvattu liitteissä 1 ja 2. Kaikki SQL-palvelimen käyttämät protokollat aktivoitiin asennuksen jälkeen liitteen 2 mukaisesti, jotta hallintapalvelin toimisi oikein. Lopuksi SQL Server-palvelu käynnistettiin uudelleen asetusten voimaan saattamiseksi.

Hallintapalvelimen sekä asiakasohjelman asennus oli yksinkertaista ja nopeaa. Kyseiset asennukset on kuvattu tarkemmin liitteissä 3 ja 4. Hallintapalvelimen asennuksen

jälkeen tarkistettiin, että Application Virtualization Management Server-palvelu oli started-tilassa. Lisäksi luotu *content*-hakemisto jaettiin lukuoikeuksin kaikille käyttäjille. Asennusten jälkeen siirryttiin varsinaiseen testausosuuteen.

Hallintapalvelimen käyttö

Hallintapalvelimen käyttö tapahtui Application Virtualization Management Consolen avulla. Palvelimen osoitteeksi laitettiin appvmanagement ja käytettäväksi portiksi 80. Palvelimeen kirjauduttiin tunnuksella, joka oli liitetty AppV_Admns-ryhmään. Management Consolen avulla tehdään käytännössä kaikki sovellusten hallintaan liittyvät toimenpiteet. Sovelluksista ja palvelimista voidaan muodostaa monia eri ryhmiä (Application Group & Server Group) hallinnan selkeyttämiseksi. Kuviosta 16 on nähtävissä Management Consolen käyttöliittymä.

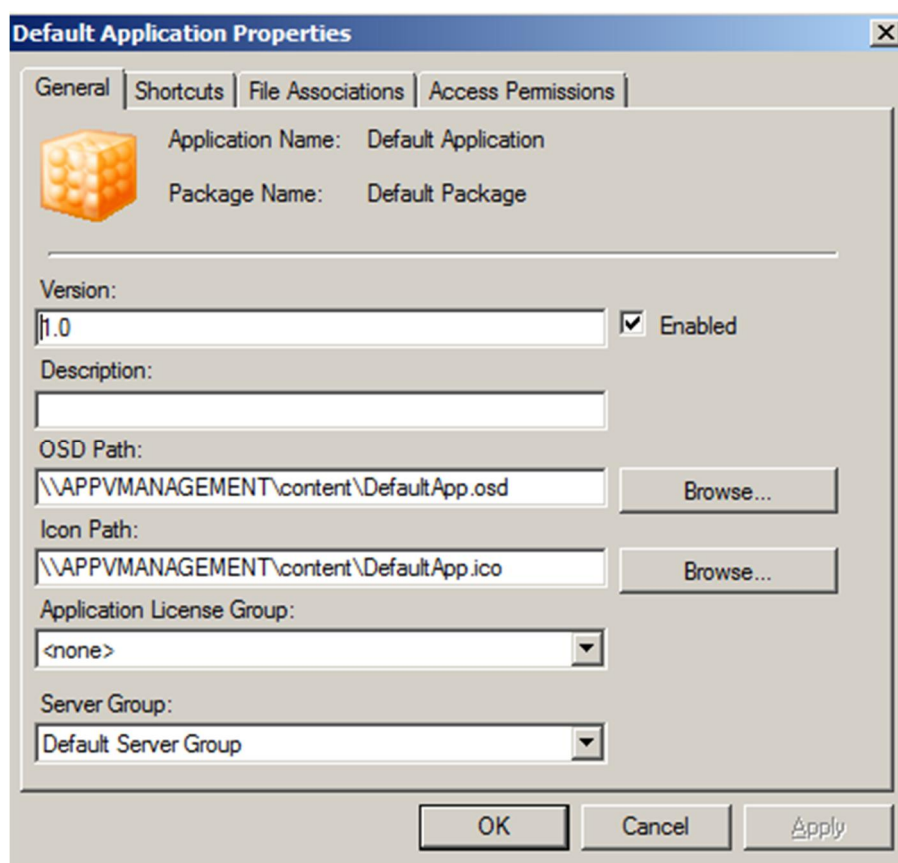


KUVIO 16. Hallintapalvelimen käyttöliittymä

Ensimmäisenä toimenpiteenä Management Consolessa klikattiin appvmanagement-palvelinta hiiren oikealla näppäimellä ja valittiin System Options. System Options:ssa

määritettiin UNC-polku (Universal Naming Convention) content-kansioon, joka oli tässä tapauksessa *//appvmanagement/content*.

Seuraavaksi kokeiltiin järjestelmän toimivuutta hallintapalvelimen mukana tulevan Default Applicationin avulla. Kyseistä sovellusta klikattiin hiiren oikealla näppäimellä ja valittiin properties. General-välilehdellä OSD Path sekä Icon Path -kenttiin määritettiin UNC-polut OSD- sekä ICO-tiedostoihin kuvion 17 mukaisesti. Access Permissions -välilehdelle lisättiin vielä AppV_Users-ryhmä, jonka avulla sovellus julkaistiin kyseiseen ryhmään kuuluville käyttäjille.

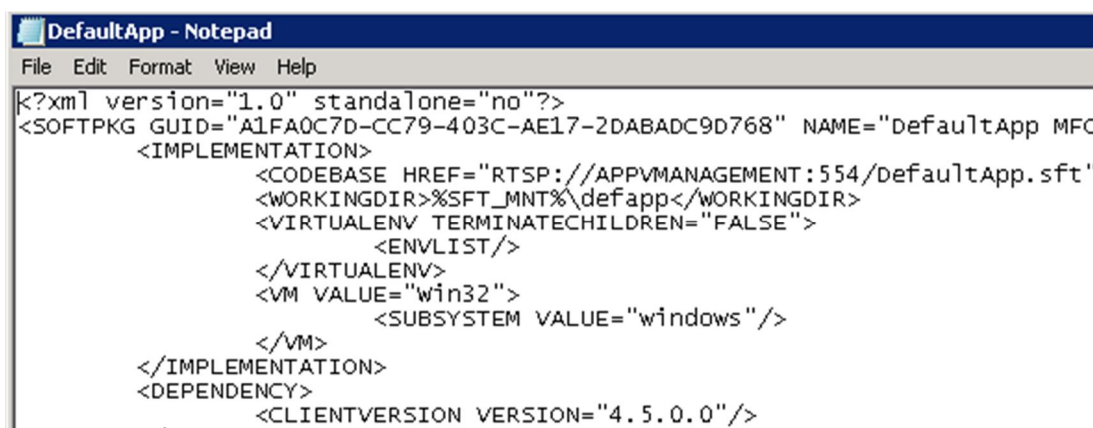


KUVIO 17. Virtualisoidun sovelluksen asetukset

Ennen siirtymistä asiakasohjelman sisältävälle päätelaitteelle tarkistettiin Default Applicationin OSD-tiedosto, joka sijaitsee content-hakemistossa. OSD-tiedostoja pystyy

tarkastelemaan ja muokkaamaan esimerkiksi muistio-sovelluksella. Kuviossa 18 on osittainen kuvakaappaus OSD-tiedoston sisällöstä. Tiedostosta tarkastettiin HREF-kentän sisältämät arvot. Arvojen piti olla

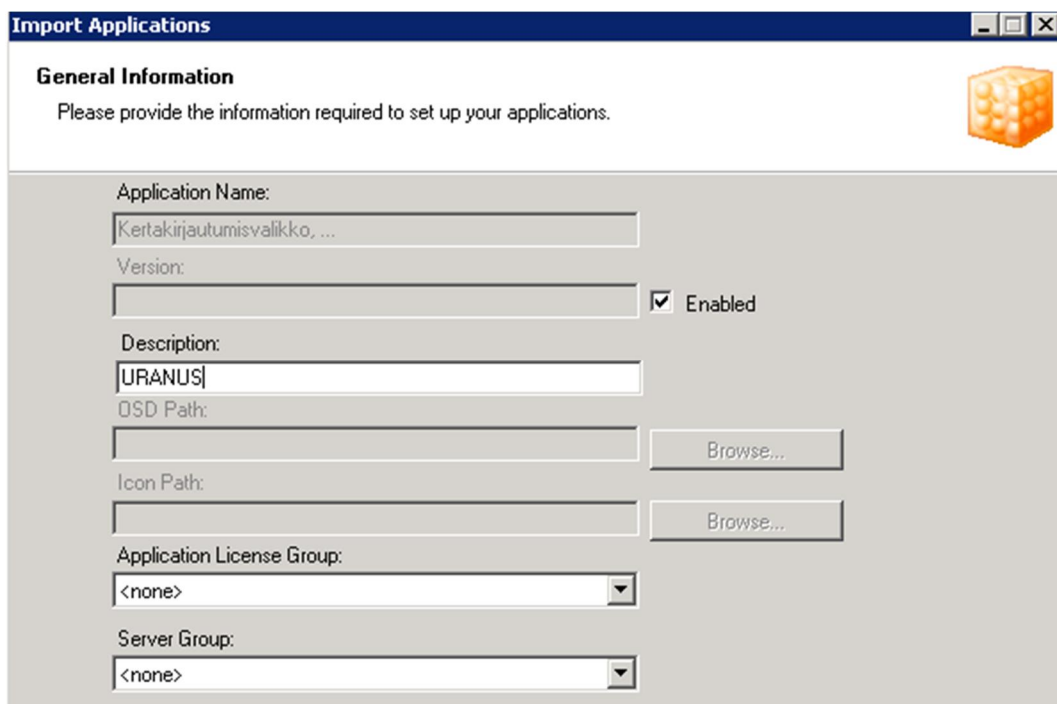
”RTSP://APPVMANAGEMENT:554/DefaultApp.sft”, jotta sovellus toimisi. RTSP on aiemmin määritetty protokolla suoratoistoa varten, APPVMANAGEMENT on hallintapalvelimen nimi, 554 on aiemmin määritetty RTSP:n kanssa käytettävä portti ja DefaultApp.sft on relatiivinen polku Default Applicationin sft-tiedostoon. Relatiivinen polku tarkoittaa rakennetta content-hakemiston jälkeen. Tässä tapauksessa relatiivinen polku oli pelkkä DefaultApp.sft, koska DefaultApp.sft sijaitsee content-hakemiston juuressa.



KUVIO 18. OSD-tiedoston osittainen sisältö

Seuraavaksi siirryttiin Windows 7 -työasemalle toiminnan varmistamiseksi. Työasemaan kirjauduttiin AppV_Users-ryhmään kuuluvalla tunnuksella. Työaseman työpöydälle oli ilmestynyt Default Application kuvake. Kuvaketta klikkaamalla työaseman oikeaan alakulmaan tuli ilmoitus sovelluksen käynnistymisestä ja hetken kuluttua sovellus oli käytettävissä. Koska kokeilu onnistui, siirryttiin takaisin hallintapalvelimelle aiemmin luodun uranus-sovelluksen liittämiseksi järjestelmään.

Aluksi sekvensseri-työasemalla oleva uranus2-hakemisto siirrettiin hallintapalvelimen content-kansioon. Siirron jälkeen jokaisen OSD-tiedoston HREF-arvo tarkistettiin aikaisemmin esitetyn esimerkin mukaisesti. Uuden sovelluksen lisääminen hallintapalvelimeen tehtiin Management Consolella. Applications-kohtaa klikattiin hiiren oikealla näppäimellä ja valittiin Import Applications. Avautuneessa valikossa siirryttiin UNC-polkua käyttämällä uranus2-hakemistoon ja valittiin uranus2.sprj-tiedosto. Ruudulle avautui asennusvelho (wizard), jonka avulla paketin julkaisuasetukset konfiguroitiin. Asennusvelhon ensimmäisellä sivulla muokattiin ainoastaan Description-kenttää. Kaikki kenttiä ei kuitenkaan voitu muokata, kuten kuvioista 19 voidaan havaita.



Import Applications

General Information
Please provide the information required to set up your applications.

Application Name:
Kertakirjautumisvalikko, ...

Version:
[Empty]

Description:
URANUS

OSD Path:
[Empty]

Icon Path:
[Empty]

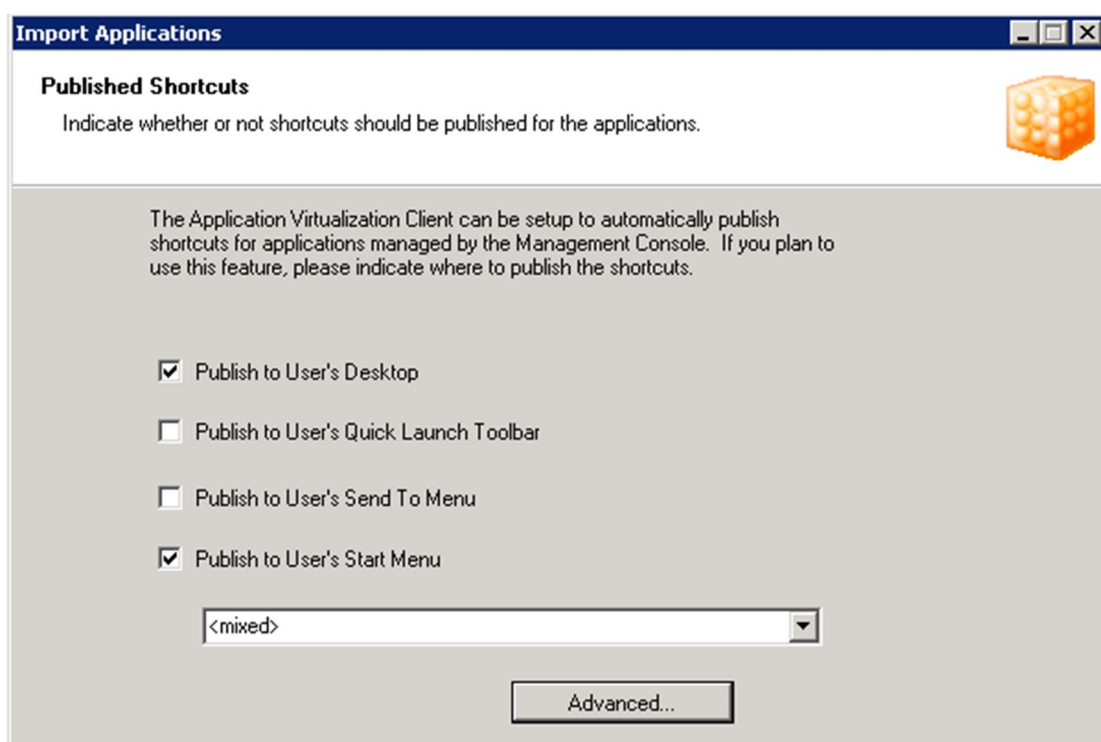
Application License Group:
<none>

Server Group:
<none>

☒ Enabled

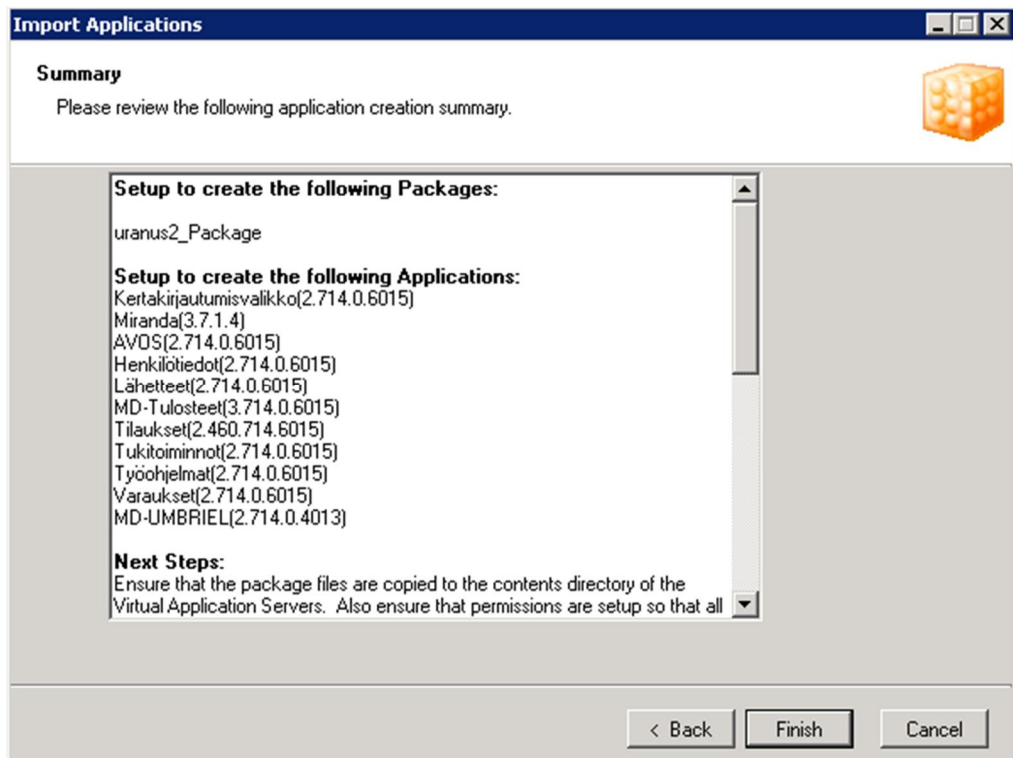
KUVIO 19. Sovelluksen lisääminen hallintapalvelimelle

Seuraavalla sivulla määritettiin pikakuvakkeiden sijainti työasemissa, joihin sovellus julkaistaan. Tässä tapauksessa pikakuvakkeet julkaistiin työpöydälle ja käynnistysvalikkoon kuvion 20 mukaisesti. Seuraavaksi määritettiin käyttöoikeudet AppV_Users-ryhmälle.



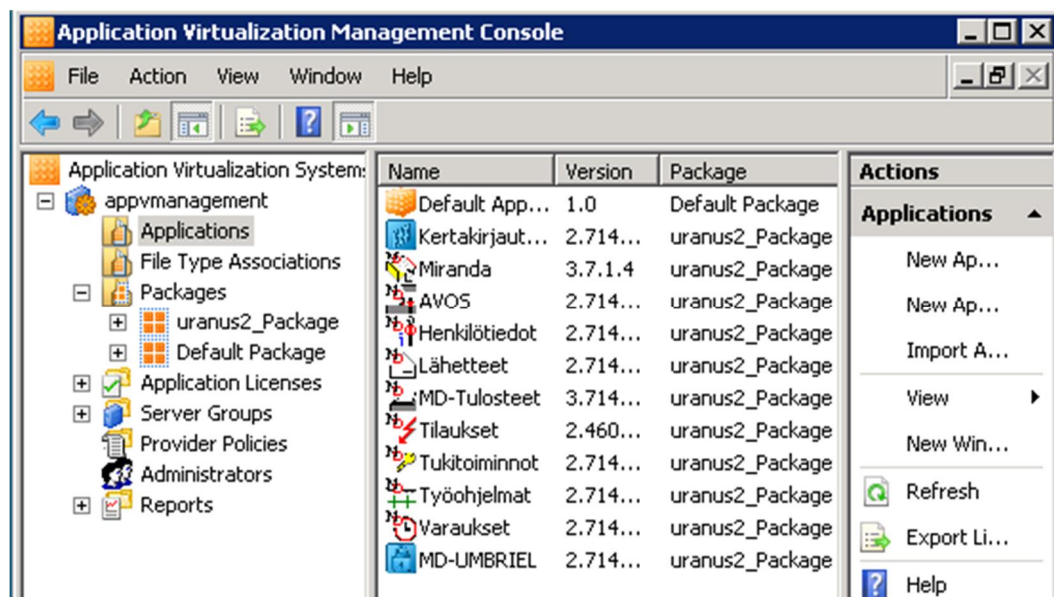
KUVIO 20. Pikakuvakkeiden sijainnin määrittäminen

Viimeisen sivun yhteenvedosta nähtiin lisättävät sovellukset, luotavan paketin nimi ja jatkotoimenpiteet. Yhteenvedon sisältämät tiedot ovat nähtävissä osittain kuviossa 21.



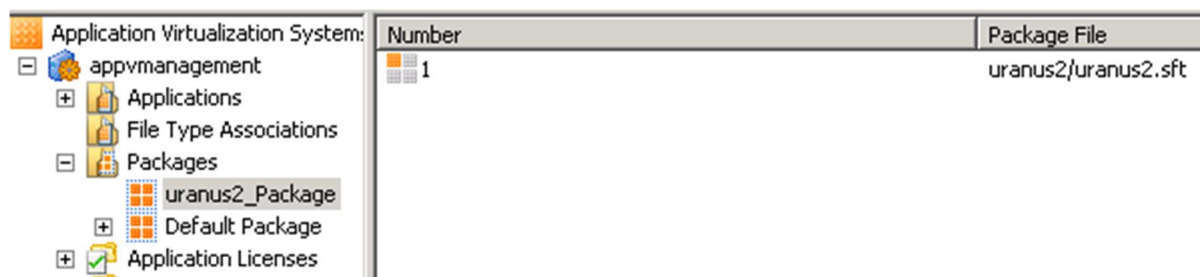
KUVIO 21. Yhteenveto sovelluksen lisäämisestä

Kuviosta 22 nähdään lisätty uranus2-paketti, sekä sen sisältämät sovellukset.



KUVIO 22. Näkymä sovellusten lisäyksen jälkeen

Seuraavaksi tarkastettiin kuvion 23 mukaisesti, että uranus2_packagen Package File -kentässä oleva relatiivinen polku oli oikein.



KUVIO 23. Paketin relatiivisen polun tarkastus

Tarkastusten jälkeen siirryttiin jälleen Windows 7 -työasemalle. Työasemalta kirjaututtiin ulos ja takaisin sisään uusien asetusten voimaansaattamiseksi. Vaihtoehtoisesti olisi voitu käynnistää Application Virtualization Client esimerkiksi komentoriviltä komennolla SftCMC.msc ja valita Publishing Server-kohdasta julkaisupalvelimen päivitys. Julkaistut sovellukset ilmestyivät työpöydälle ja käynnistä-valikkoon niin kuin oli määritetty.

Suoratoiston testaus

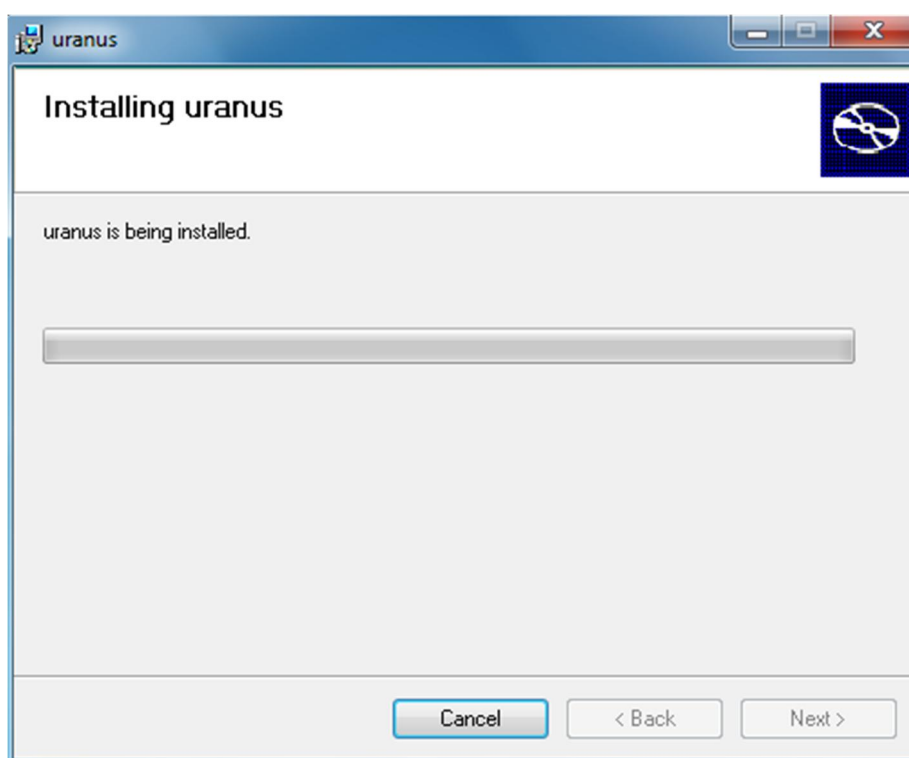
Sovellusten ensimmäinen käynnistys kesti muutamia sekunteja, koska sovelluksen käynnistyslohko FB1 piti ladata kokonaisuudessaan työasemalle. Sovelluksen eri toimintoja käytettiin vähän aikaa ennen sen sulkemista. Seuraavaksi sama sovellus käynnistettiin uudestaan. Tällöin sovellus käynnistyi saman tien, koska sovelluksen käynnistyslohko oli jo ladattu välimuistiin.

5.3.3 Itsenäinen malli

Itsenäisen mallin testaaminen suoritettiin aiemmin luodulla Uranus-sovelluksella. Aluksi toiselle Windows 7 -työasemalle asennettiin asiakasohjelmisto samaan tapaan kuin ensimmäisellekin. Asennuksen jälkeen asiakasohjelmiston rekisteriarvoja täytyi

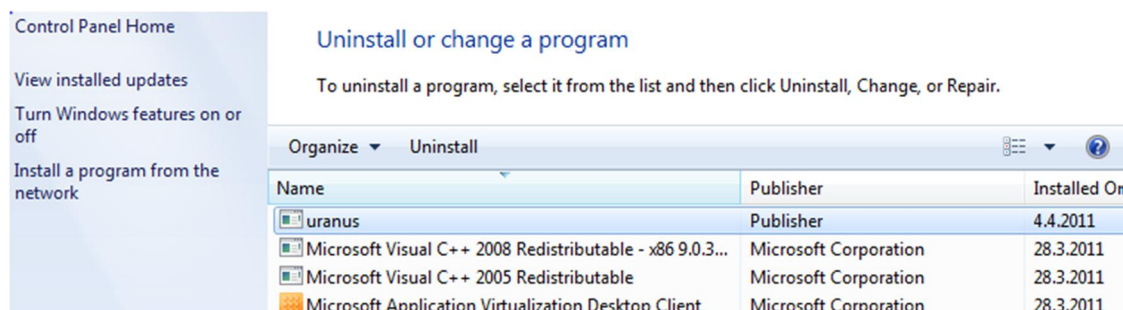
muuttaa, jotta itsenäistä mallia voitiin käyttää. Aluksi komentorivillä kirjoitettiin komento *regedit*, jotta rekisterieditoriin päästiin käsiksi. Rekisterieditorissa siirryttiin *HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\SoftGrid\4.5\Client\Configuration* -hakemistoon. Hakemistosta valittiin *RequireAuthorizationIfCached* ja Value data-kentän arvoksi määritettiin 0. Seuraavaksi siirryttiin samassa puurakenteessa sijaitsevaan Network-hakemistoon, jossa muokattiin *AllowDisconnectedOperation*- ja *Online*-rekisteriarvoja. Ensimmäisenä mainitun arvoksi määritettiin 1, kun taas jälkimmäisen arvon täytyi olla 0. Viimeinen rekisterimuutos tehtiin samassa puurakenteessa sijaitsevassa Permissions-hakemistossa, jossa *ToggleOfflineMode*:n rekisteriarvoksi määritettiin 0. Sen jälkeen rekisterieditori suljettiin ja *Application Virtualization Client*-palvelu käynnistettiin uudelleen Windows services-hallintakonsolin avulla.

Seuraavaksi hallintapalvelimella uranus2-hakemistossa olleet uranus.msi- ja uranus.sft-tiedostot siirrettiin samaan kansioon Windows 7 -työasemalle. Sovellus asennettiin MSI-paketin avulla työasemaan samaan tapaan kuin mikä tahansa perinteisesti asennettava sovellus, kuten kuvioista 24 voidaan havaita.



KUVIO 24. Virtualisoidun sovelluksen asennus MSI-paketilla

MSI-paketilla asennettu sovellus rekisteröityy käyttöjärjestelmään. Siten se voidaan myös poistaa samaan tapaan kuin perinteinen sovellus, kuten kuvioista 25 voidaan havaita.



KUVIO 25. MSI-paketilla asennetun virtuaalisovelluksen poistaminen

MSI-paketti loi pikakuvakkeet työasemalle määritysten mukaan. Seuraavaksi varmistettiin, että sovellus käynnistyy luoduista pikakuvakkeista. Sovelluksen käynnistyminen voitiin havaita työpöydän oikeassa alakulmassa näkyvästä ilmoituksesta. Sovellus käynnistyi siis samaan tapaan kuin suoratoistettu sovellus. Itsenäisen mallin tapauksessa sovellus käynnistyi hieman nopeammin, koska sovellusta ei tarvinnut ladata verkon yli välimuistiin.

5.3.4 Sovellusten linkitys

Sovellusten linkitystä testattiin virtualisoitujen Firefox- ja Foxit Reader-sovellusten välillä. Firefox virtualisoitiin tavalliseen tapaan, kun taas Foxit Reader virtualisointi poikkesi hieman tästä. Ennen sekvensserin käynnistystä käyttöjärjestelmään täytyi asentaa Firefox. Muuten virtualisointi tehtiin perinteisellä tavalla.

Luodut sovellukset siirrettiin Appvmanagement-palvelimelle content-hakemistoon. Firefox.osd-tiedostoa muokattiin linkityksen muodostamiseksi. Linkitys tehtiin siten, että Foxit.osd-tiedoston <CODEBASE HREF-alkuinen rivi kopioitiin kokonaisuudes-

saan Firefox.osd-tiedostoon <DEPENDENCIES>-tekstien väliin kuvion 26 mukaisesti.

```
</POLICIES>
<ENVLIST/>
<DEPENDENCIES>
  <CODEBASE HREF="RTSP://appvmanagement:554/foxit/foxit.sft" GUID="DC4F6C55-0A1D-
</DEPENDENCIES>
```

KUVIO 26. Linkityksen muodostus

Molemmat sovellukset lisättiin hallintapalvelimelle Import-komennolla. Seuraavaksi siirryttiin Windows 7 -työasemalle kokeilemaan linkin toimivuutta. Työasemalta käynnistettiin Firefox ja siirryttiin osoitteeseen, josta PDF-tiedosto avautuisi suoraan. PDF-tiedosto avautui selaimeen määritysten mukaan, joten luotu linkki havaittiin toimivaksi.

5.4 ThinApp

Virtualisoituja sovelluksia testattiin molemmilla toimintamalleilla. Streamed Execution Mode:n toimivuutta testattiin sijoittamalla sovellukset appvmanagement-palvelimella olevaan jaettuun hakemistoon, johon oli asetettu lukuoikeus kaikille käyttäjille. Deployed Execution Mode:a testattiin tallentamalla luotu sovellus käyttäjän työaseman kovalevylle sekä USB-muistitikulle. USB-muistitikulta käynnistettävä sovellus oli virtualisoitu uudestaan siten, että hiekkalaatikon sijainti oli samassa hakemistossa sovelluksen kanssa. Testauksessa kiinnitettiin huomiota mallien toimintakuntoon saattamisen vaativuuteen, käytettävyyteen ja virtualisoitujen sovellusten toimintaan.

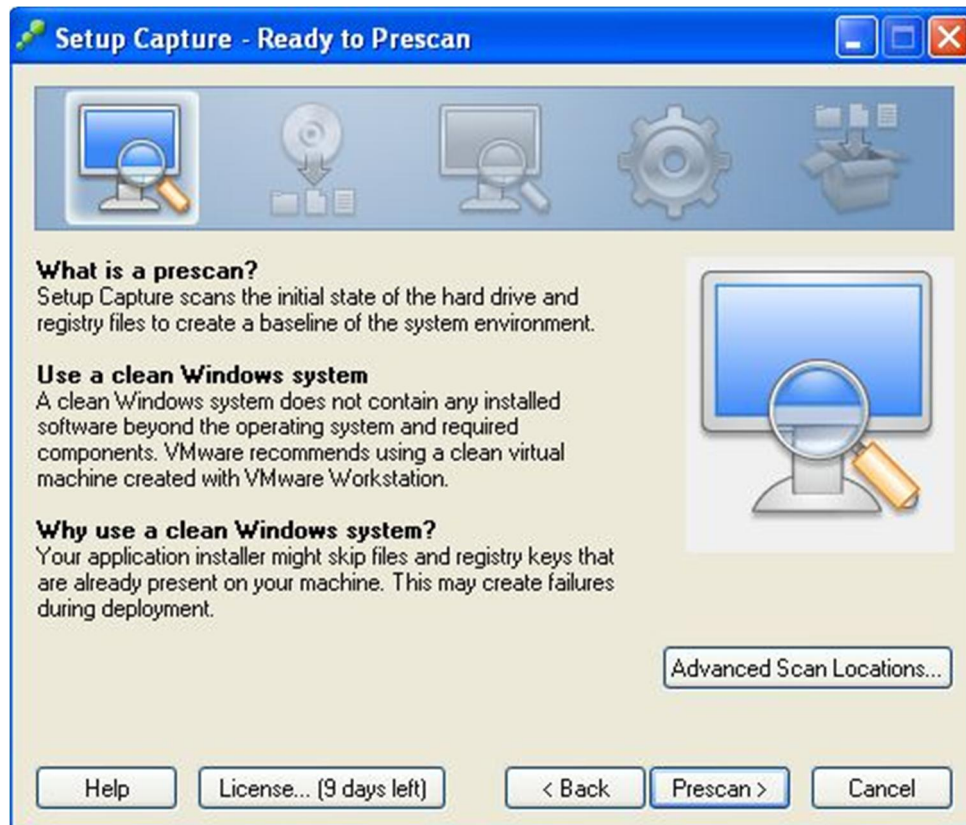
5.4.1 Setup Capturen käyttö

Setup Capturen ensimmäiseltä sivulta, joka on nähtävissä kuvioista 27, voitiin havaita sovelluksen virtualisointiprosessin sisältämät vaiheet ja käytettävä ThinApp:n versio.



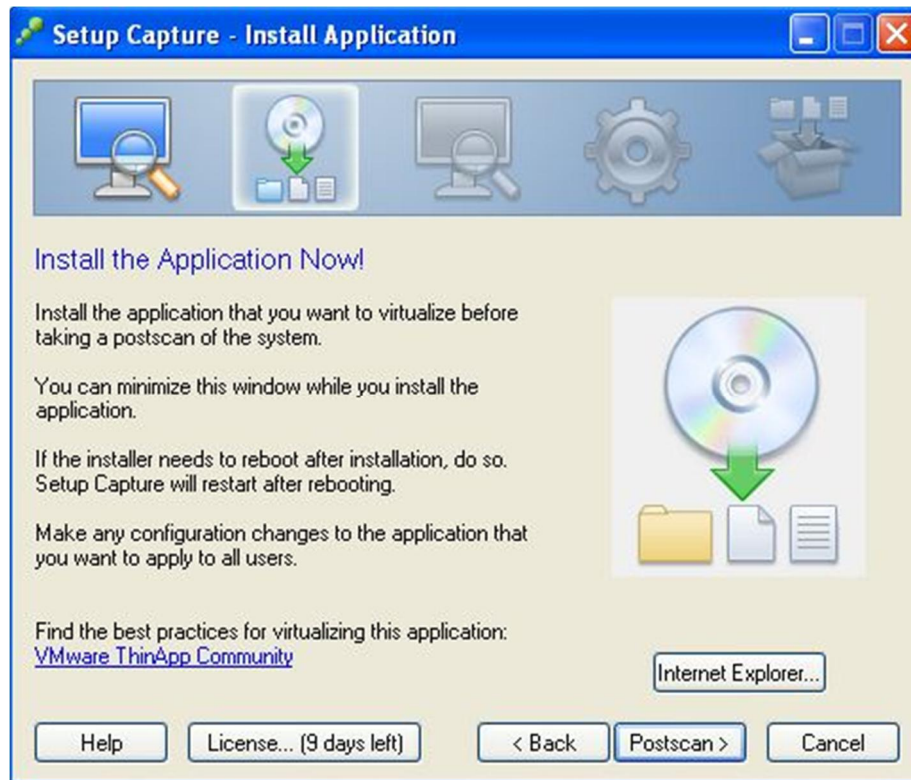
KUVIO 27. Setup Capture-prosessi kokonaisuudessaan

Virtualisointiprosessi alkoi varsinaisesti seuraavalta sivulta. Sivulta aloitettiin käyttöjärjestelmän skannaus *Prescan*-painikkeesta vertailukohdan muodostamiseksi. Skannaus suoritettiin oletusasetuksilla. Skannaus-sivulla kerrotaan lyhyesti, mitä Prescan tarkoittaa ja minkälaisilla asetuksilla skannaus kannattaa suorittaa. Skannaus-sivu on nähtävissä kuvioista 28.



KUVIO 28. Järjestelmän skannaus vertailukohdan muodostamiseksi

Skannauksen jälkeen oli vuorossa virtualisoitavan sovelluksen asennus. Setup Capture antoi selvät ohjeet sovelluksen asennukseen ja jatkotoimenpiteisiin. Sovelluksen asennuksen jälkeen klikattiin *Postscan*-painiketta eroavaisuuksien skannaamiseksi ja lopulta virtualisoidun sovelluksen muodostamiseksi. Kuvioista 29 nähdään edellä mainittujen toimenpiteiden lisäksi myös painike ”Internet Explorer..”. Painike mahdollistaa esimerkiksi Internet Explorer 6:n virtualisoinnin, kun Setup Capture:a käytetään Windows XP -käyttöjärjestelmässä.



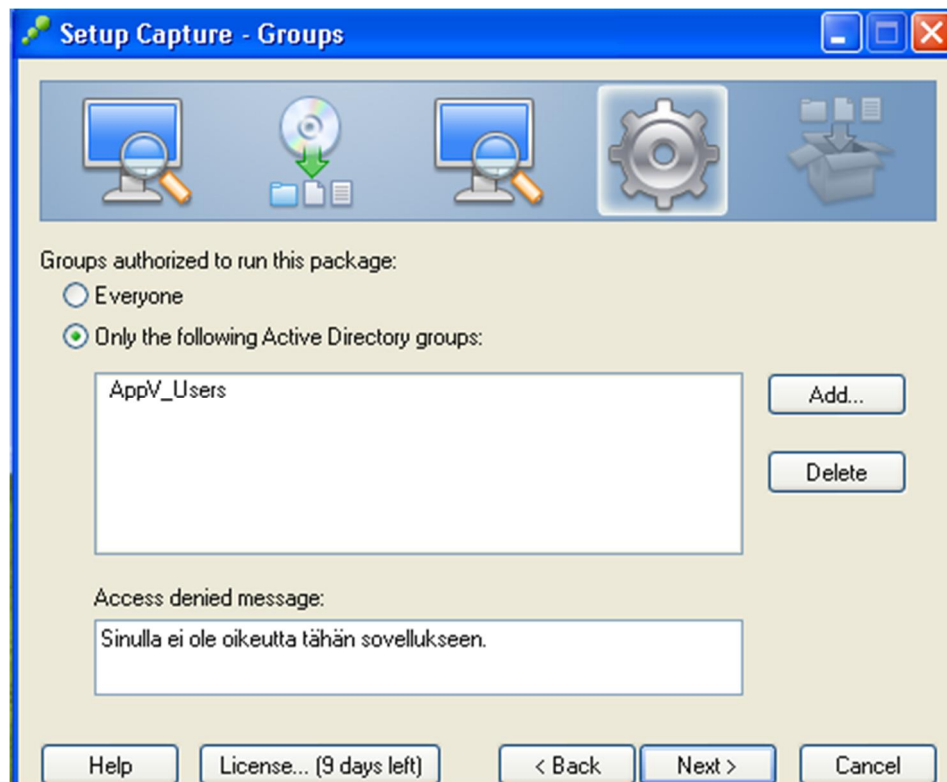
KUVIO 29. Sovelluksen asennus & postscan

Kuviossa 30 on Entry Point-sivu, joka sisältää skannauksen perusteella luodut sovellukset. Sivulta valittiin käytettävät aloituspisteet. Tutkimuksessa käytettiin Setup Capturen ehdottamia aloituspisteitä.



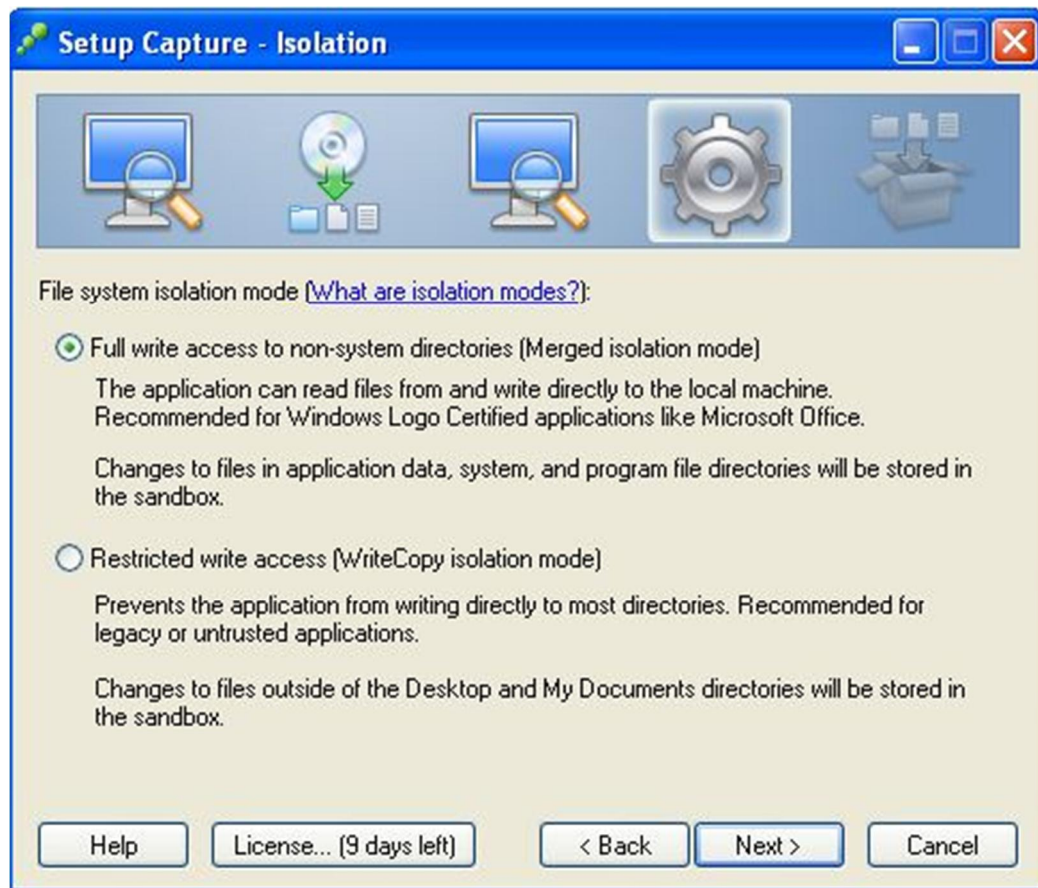
KUVIO 30. Aloituspisteiden valinta

Aloituspisteiden määrittämisen jälkeen valittiin käyttäjät, joilla on oikeus käyttää sovellusta. Oikeudet määritettiin aiemmin luodulle AppV_Users-ryhmälle kuvion 31 mukaisesti. Lisäksi määritettiin näytettäväksi viesti: ”Sinulla ei ole oikeutta tähän sovellukseen” niiden käyttäjien varalle, jotka yrittävät käyttää sovellusta ilman oikeuksia.



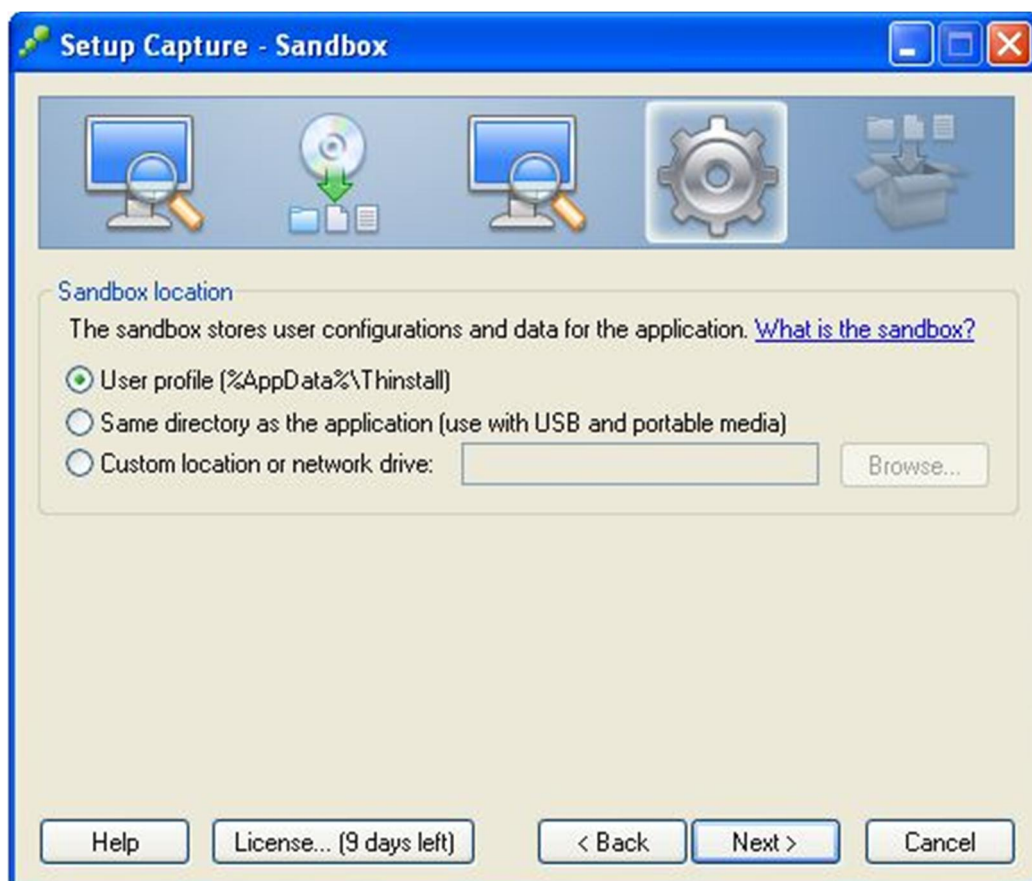
KUVIO 31. Käyttöoikeuksien määrittäminen

Kuviossa 32 nähdään eristystilamäärittäykset, niiden kuvaukset ja käytetty eristystila, joka oli tässä tapauksessa yhdistetty (merged isolation mode).



KUVIO 32. Eristystilan määrittäminen

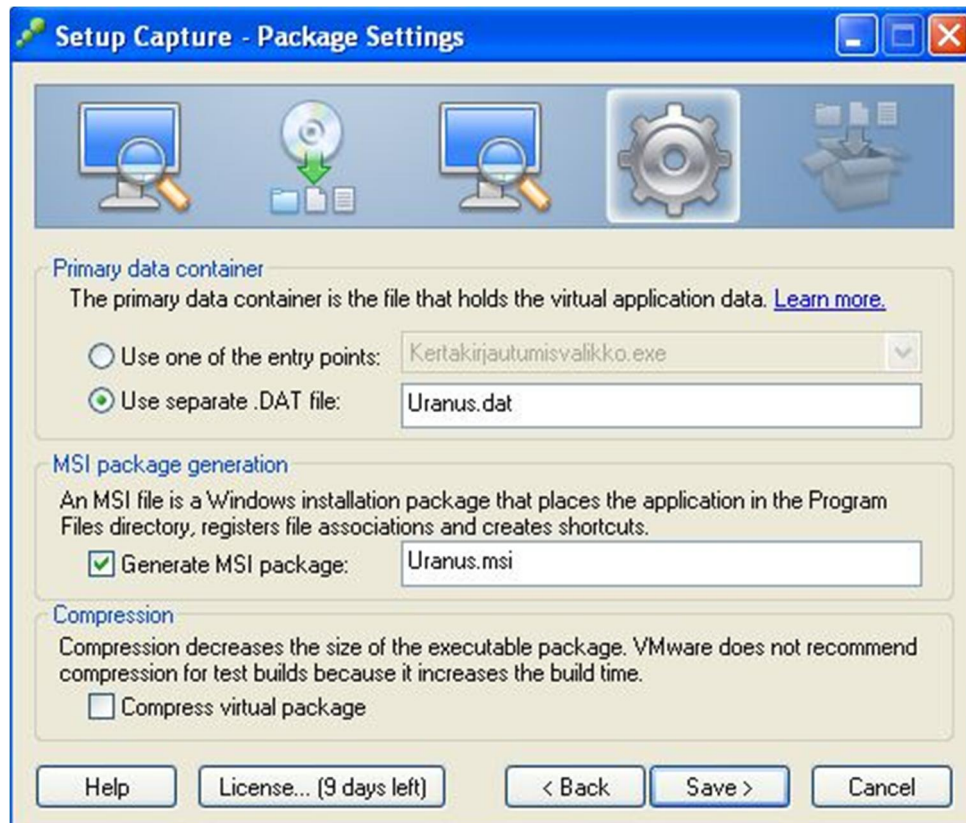
Seuraavaksi määritettiin hiekkalaatikon sijainti. Sijainniksi valittiin Setup Capturen ehdottama käyttäjän profiili (%AppData%\Thinstall) kuvion 33 mukaisesti.



KUVIO 33. Hiekkalaatikon sijainnin määrittäminen

Hiekkalaatikon sijainnin määrittämisen jälkeen täytyi valita, lähetetäänkö luodun paketin tietoja VMwarelle. Tässä tapauksessa tietojen lähetystä ei sallittu. Seuraavaksi luotavalle paketille täytyi antaa nimi ja valita projektin tallennuspaikka. Setup Capturen ehdottama nimi oli varsin pitkä, joten se lyhennettiin muotoon Uranus. Projektin tallennuspaikaksi valittiin Setup Capturen ehdottama oletuskansio.

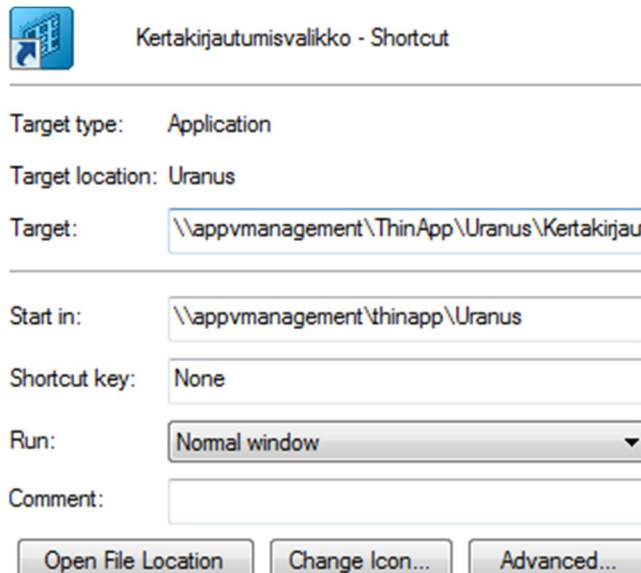
Viimeisenä vaiheena määritettiin paketin sisältämän tiedon säilytyspaikka (Primary data container) sekä MSI-paketin luonti. Koska luotava paketti oli varsin suuri ja se sisälsi lukuisia sovelluksia, tiedon säilytyspaikaksi valittiin erillinen .DAT-tiedosto Uranus.dat kuvion 34 mukaisesti. Luotava paketti olisi voitu myös pakata tilan säästämiseksi, mutta sitä ei tehty tässä tutkimuksessa.



KUVIO 34. Paketoinnin viimeinen vaihe

5.4.2 Streamed Execution Mode

Streamed Execution Mode:n testaus oli hyvin yksinkertaista. Appvmanagement-palvelimelle luotiin hakemisto ThinApp, joka jaettiin lukuoikeuksilla kaikille käyttäjille. Koko bin-kansio siirrettiin kyseiseen hakemistoon, koska Uranus loi asentueksaan lukuisia eri sovelluksia. Oleellista oli, että virtualisoitu sovellus sekä Uranus.dat-tiedosto olivat samassa hakemistossa. Bin-hakemiston nimeksi vaihdettiin vielä Uranus. Sovelluksista luotiin pikakuvakkeet, jotka osoittivat kohteeseen `\\appvmanagement\ThinApp\Uranus\Sovellus`. Kuviossa 35 on esimerkki luodun pikakuvakkeen polusta.



KUVIO 35. Pikakuvakkeen polku

Seuraavaksi pikakuvake kopioitiin Windows 7 -työasemalle. Työasemalle kirjaututtiin AppV_Users-ryhmään kuuluvalla tunnuksella ja käynnistettiin sovellus pikakuvakkeesta. Sovellus käynnistyi ongelmitta heti käynnistyslohkojen latauksen jälkeen.

Seuraavaksi työasemalta kirjaututtiin ulos ja takaisin sisään käyttäjällä, joka ei kuullut edellä mainittuun ryhmään. Kun sovellusta yritettiin käynnistää, ruutuun ilmestyi aiemmin määritetty teksti: ”Sinulla ei ole oikeutta tähän sovellukseen”. Käyttäjäoikeuksien määritykset olivat siis voimassa.

5.4.3 Deployed Execution Mode

Deployment Execution Mode:n testaus aloitettiin Uranus.dat- sekä Uranus.msi-tiedostojen siirrolla Windows 7-työasemalle samaan hakemistoon. Seuraavaksi virtualisoitu sovellus asennettiin käyttöjärjestelmään Uranus.msi-tiedoston avulla, joka loi pikakuvakkeet ja rekisteröi sovelluksen käyttöjärjestelmään. Seuraavaksi sovellus käynnistettiin työpöydälle ilmestyneestä pikakuvakkeesta. Sovellus käynnistyi odotetusti muutamassa sekunnissa. Käyttöoikeuksia ei testattu muilla käyttäjillä edellä ole-

van testauksen vuoksi. Sovelluksen poisto oli mahdollista tehdä hallintapaneelin kautta samalla tavalla kuin App-V:n tapauksessa.

Seuraavaksi testattiin sovellusten toimivuutta USB-muistitikkua käyttämällä. USB-muistitikulle siirrettiin koko bin-hakemisto. Uranus oli paketoitu uudestaan siten, että hiekkalaatikon sijainniksi oli valittu sama hakemisto kuin itse sovelluksella. Tällä ratkaisulla saavutettiin se, että sovelluksen asetukset ja tiedot pysyivät aina mukana, käytettiinpä mitä konetta tahansa. Sovellukset toimivat muistitikkua käyttämällä samaan tapaan kuin paikalliselta levyltä käytettynä.

5.4.4 Sovellusten linkitys

Sovellusten linkitystä testattiin virtualisoitujen Firefox- ja Foxit Reader-sovellusten välillä. Foxit Reader siirrettiin Appvmanagement-palvelimelle aiemmin luotuun jaettuun ThinApp-hakemistoon ja sovelluksen nimi muutettiin muotoon Foxit.exe. Seuraavaksi Firefox:n package.ini-tiedostoa muokattiin linkityksen muodostamiseksi. Initiedostossa siirryttiin riville *;OptionalAppLinks*. Rivi muutettiin muotoon *OptionalApplinks=\\appvmanagement\ThinApp\Foxit.exe*. Foxit Reader oli konfiguroitu siten, että PDF-tiedostojen pitäisi avautua suoraan selaimeen. Firefox-sovelluksella siirryttiin osoitteeseen, joka avaisi PDF-tiedoston. PDF-tiedosto avautui tarkoituksen mukaisesti suoraan selaimeen.

6 TESTAUKSEN TULOKSET

6.1 App-V

App-V-ympäristö

App-V-ympäristön konfigurointi ja käyttäminen oli pääosin sujuvaa. Ympäristön rakentaminen kesti muutamia tunteja hallintapalvelimen ja sen vaatimien komponenttien vuoksi. Ympäristön rakentamisen ja konfiguroinnin aikana tuli selväksi, kuinka tärkeää ohjeiden lukeminen ja noudattaminen on järjestelmän toiminnan kannalta.

Konfigurointivaiheen aikana vastaan tuli muutamia ongelmia, mutta niihin kaikkiin löytyi vastaus Microsoftin sivuja selaamalla. Ensimmäinen vastaan tullut ongelma liittyi ensimmäisenä luodun sovelluksen julkaisemiseen. Asiakasohjelmisto ei löytänyt sovellusta, koska OSD-tiedoston HREF-kentässä oleva polku oli kirjoitettu väärin. Olin ymmärtänyt väärin relatiivisen polun merkitsemisen. Sama relatiiviseen polun määrittelyongelma piti korjata myös Management Consolessa aiemmin mainitun kuvion 22 tapaisesti. Myöhemmin vastaan tuli ongelma, jossa Management Consoleen ei päässyt käsiksi. Syyksi paljastui Application Virtualization Management Service-palvelun väärä tila, joka korjattiin vaihtamalla tilaksi *started*. Muita merkittäviä ongelmia järjestelmän käytössä tai sovellusten julkaisemisessa ei ilmennyt.

Sovellusten hallinta ja julkaiseminen käyttäjille onnistui vaivattomasti Management Consolea käyttämällä. Aktiivihakemiston ryhmien avulla sovellukset saatiin helposti kohdistettua halutuille käyttäjille. Sovelluksia oli vaivatonta lisätä, poistaa tai jakaa erillisiin ryhmiin selkeän käyttöliittymän ansiosta.

Virtualisoidut sovellukset

Sovellusten virtualisointi ja konfigurointi oli yksinkertaisten sovellusten kohdalla helppoa, sillä muutoksia ei tarvinnut juuri tehdä. Saman sovelluksen virtualisoinnin voi tehdä uudelleen ilman ohjeita. Suurien ohjelmistokokonaisuuksien, kuten Office 2010:n virtualisointi oli huomattavasti haastavampaa. Office 2010:n tapauksessa käyttöjärjestelmään täytyi tehdä lukuisia muutoksia ennen asennusta. Ilman tarkkoja ohjeita virtualisointi olisi tuskin onnistunut. Testatut sovellukset toimivat pääosin halutulla tavalla, sillä ainoastaan Uranuksen kaikkia haluttuja ominaisuuksia ei saatu toimimaan.

Virtualisoitujen sovellusten välisen linkin luominen oli yllättävän nopeaa ja helppoa, sillä se onnistui heti ensimmäisellä yrityksellä. Virtualisoidun sovelluksen kokoa saatiin vähennettyä Uranuksen tapauksessa yli 50 % pakkausta käyttämällä, jolloin verkkoliikenteen määrä saadaan puolitettua pakkauksen ansiosta. Pakkaus kuitenkin aiheuttaa sen, että asiakasohjelman sisältämä työasema joutuu purkamaan tiedon ennen käyttöä. Kyseinen operaatio kuormittaa jonkin verran työaseman prosessoria. Tehottomissa koneissa se voi tuottaa ongelmia, joten sen käyttö ei ole aina suotavaa. Kuor-

mituksen määrää testattiin kannettavalla tietokoneella, joka sisälsi Intel Core 2 Duo P9400-prosessorin ja kolme gigatavua muistia. Kyseistä tietokonetta käytettäessä pakkauksen vaikutus koneen toimintaan ei ollut havaittavissa.

Virtualisoituja sovelluksia käytettäessä käyttökokemus oli hyvin lähellä perinteisiä sovelluksia. Sovellukset käynnistyivät nopeasti lukuun ottamatta muutamia kertoja, jolloin sovellus tarkisti päivityksiä muutamia sekunteja. Loppukäyttäjä tuskin erottaisi, onko käytössä virtualisoitu vai perinteisesti asennettu sovellus ilman työpöydän oikeassa alakulmassa näkyvää palkkia, joka ilmoittaa sovelluksen käynnistymisestä. Palkin poistamiseen löytyi ohjeita, mutta poistamista ei toteutettu tällä kertaa.

6.2 ThinApp

ThinApp ei vaatinut erillisen ympäristön pystyttämistä, joten App-V:n kaltaisia ympäristöön liittyviä ongelmia ei ollut. ThinApp:lla yksinkertaisten sovellusten virtualisointi oli jopa helpompaa kuin App-V:llä. Setup Capturen aikana tulleet ohjeet riittivät virtualisoinnin toteuttamiseen yksinkertaisten sovellusten kohdalla, joten erillisiä ohjeita ei tarvinnut lukea samalla. Suurien ohjelmistokokonaisuuksien (esimerkiksi Office 2010) virtualisointi oli App-V:n tapaan huomattavasti haastavampaa. Virtualisoinnissa täytyi tehdä lukuisia eri muutoksia ennen asennusta, asennuksen aikana ja asennuksen jälkeen, joten ilman yksityiskohtaisia ohjeita virtualisointi tuskin olisi onnistunut. Virtualisoidut sovellukset toimivat hyvin Uranusta lukuun ottamatta, jonka kanssa oli samoja ongelmia kuin App-V:llä.

Virtualisoitujen sovellusten välisen linkin luonti oli helppoa ja tässäkin tapauksessa linkitys onnistui ensimmäisellä yrityksellä. Virtualisoidun sovelluksen kokoa saatiin vähennettyä esimerkiksi Uranus-sovelluksen kohdalla yli 30 % pakkausta käyttämällä. Pakkauksen vaikutusta sovelluksen käynnistymisnopeuteen ei voinut havaita käytettäessä aiemmin mainittua kannettavaa tietokonetta. Pakkauksen käyttöä kannattaa siis hyödyntää varsinkin silloin, kun käytettävä kaista on rajoittavana tekijänä.

Käyttökokemus oli App-V:n tapaan hyvin lähellä perinteisiä sovelluksia. Sovellukset käynnistyivät nopeasti niin muistitikulta kuin suoratoistettuna. Loppukäyttäjä tuskin

erottaisi, onko käytössä virtualisoitu vai perinteisesti asennettu sovellus ilman työpöydän oikeassa alakulmassa näkyvää palkkia, joka ilmoittaa sovelluksen käynnistymisestä. Palkin poistamiseen löytyi ohjeita, mutta poistamista ei toteutettu tällä kertaa.

7 POHDINTA

Opinnäytetyön tekeminen alkoi sovellusvirtualisointiin sekä Microsoft App-V- ja VMware ThinApp -virtualisointitekniikoihin perehtymisellä. Aineiston kartoituksessa lähdettiin liikkeelle aihetta käsittelevästä perustutkimuksesta, sillä aihe oli tutkijalle hieman vieras. Tutkimuksella pyrittiin selvittämään, miten sovellusvirtualisointi toteutetaan valituilla tekniikoilla, sekä kuinka toimivia ratkaisuja ne ovat. Sovellusten virtualisointia testattiin käytännössä edellä mainituilla sovellusvirtualisointitekniikoilla. Tutkimusongelmiin saatiin vastaus teoksista ja muista lähteistä saatujen tietojen sekä käytännön testauksien avulla.

Tutkimuksen luonteen mukaisesti saatiin laadittua dokumentti sovellusvirtualisoinnista ja VMware ThinApp- ja Microsoft App-V -tekniikoiden ominaisuuksista. Dokumentissa käsiteltiin tekniikoiden vaatimia asennuksia sekä valittujen sovellusten virtualisointia ja testausta kyseisillä tekniikoilla.

Työssä käytetyt sovellusvirtualisointitekniikat antoivat hyvän kuvan kahdesta erilaisesta ratkaisuvaihtoehdosta sekä niiden kyvyistä ja haittapuolista. App-V:n tapauksessa virtualisoituja sovelluksia oli helppo hallita yhdestä paikasta hallintapalvelimen sisältämän Management Consolen avulla. Vaihtoehtoisesti App-V:n voi myös liittää SCCM-järjestelmään, jolloin tarvetta uuden hallintajärjestelmän opetteluun ei ole. ThinAppilla virtualisoituja sovelluksia hallitaan olemassa olevilla hallintajärjestelmillä tai aktiivihakemiston ryhmäkäytännöillä, jolloin tarvetta uuden hallintajärjestelmän opettelemiselle ei ole. Asennuksessa tuli siis ainoastaan sisäistää virtualisointiprosessi.

Sovellusvirtualisointitekniikoiden omaksuminen oli nopeaa selkeän graafisen käyttöliittymän ansiosta. Sovellusten virtualisointi onnistui pääosin halutulla tavalla molemmilla tekniikoilla, joten myös käytännön osuutta voidaan pitää onnistuneena. Erot

virtualisoitujen sovellusten käytettävyydessä tai hallittavuudessa olivat loppujen lopuksi hyvin pieniä. Merkittävin ero virtualisointitekniikoiden välillä on havaittavissa niiden toteutustavassa: App-V on agentillinen ratkaisu, kun taas ThinApp on agentiton. Tutkimuksesta saamieni tuloksien perusteella voin suositella molempia tekniikoita käytettäväksi sovellusten virtualisoinnissa.

Saatuja tuloksia voidaan pitää osittain luotettavina, sillä testit suoritettiin käyttämällä samanlaisia virtuaalisia ja fyysisiä työasemia samassa ympäristössä. Suoratoiston testauksessa virtualisoidut sovellukset olivat samalla palvelimella, jolloin työaseman ja suoratoistettavan sovelluksen välimatka oli sama molemmilla tekniikoilla. Saadut tulokset perustuvat vain yhdellä samanaikaisella käyttäjällä tehtyihin testeihin, joten sovellusvirtualisointiratkaisujen toimivuutta suuremmilla käyttäjämäärillä ei voida tietää saatujen tulosten perusteella.

Työssä käytetyt sovellusvirtualisointitekniikat ovat suosittuja, joten yleisten sovellusten virtualisointiin löytyi runsaasti ohjeita sekä alan kirjallisuudesta että internetistä. Harvinaisten sovellusten, kuten työssä käytetyn Uranus-potilastietojärjestelmän virtualisoinnista kirjallisuutta ei luonnollisestikaan ollut tarjolla. Siksi toimimattomien sovellusten tai sovellusten osien korjaaminen oli vaikeaa. Ongelmatapauksia selvittämisessä varsinkin suuria ohjelmistokokonaisuuksia käytettäessä on väistämättä tehtävä yhteistyötä sovellus- ja/tai virtualisointiratkaisujen toimittajien kanssa.

Mielestäni sovellusvirtualisoinnin suurimmat hyödyt ovat käyttöjärjestelmän koskemattomuus ja suoratoistamisen mahdollistama keskitetty hallinta, jonka avulla sovellukset voidaan muun muassa päivittää ilman käyttökatkoksia. Tässä tutkimuksessa käytetyt sovellusvirtualisointitekniikat eivät myöskään vaadi loppukäyttäjältä minäkäänlaisia lisätoimenpiteitä, joten erillistä koulutusta sovellusten käyttämiseen ei tarvita.

Sovellusvirtualisointi on kasvattanut suosiotaan ja tutkimuksesta saatujen tulosten perusteella sen voidaan arvioida olevan suosittu tekniikka myös tulevaisuudessa. Suurin este sovellusvirtualisoinnin yleistymiselle on se, että sovellustoimittajat eivät hy-

väksy sovellustensa virtualisointia. Jos sovellustoimittajat saataisiin tukemaan sovellusvirtualisointia, sen käyttämistä voitaisiin pohtia myös kriittisien sovellusten parissa.

Tutkimuksen tarkoituksena oli esitellä kahta sovellusvirtualisointitekniikkaa, ThinAppia ja App-V:tä. Tutkimuksen tarkoituksena ei siten ollut selvittää, mikä tutkimuksessa esitetyistä virtualisointitekniikoista toimeksiantajan kannattaisi valita. Mikäli toimeksiantaja päätyy käyttämään toista työssä mainittua tekniikkaa, olisi seuraavana vuorossa valitun tekniikan käyttöönoton suunnittelu ja toteutus esimerkiksi 100 käyttäjälle. Tämä sopisi hyvin myös opinnäytetyön aiheeksi.

LÄHTEET

Alvarez, A. 2011. Getting Started with Microsoft Application Virtualization 4.6. Birmingham: Packt Publishing.

Bansal, J. 2010. Application Virtualization Survey Reveals Hesitation. Viitattu 26.2.2011. <http://www.appdynamics.com/blog/2010/09/01/application-virtualization-survey>.

Brodkin, J. 2010. Enterprises planning Windows 7 upgrade in 2011. Uutinen. Viitattu 8.2.2011. <http://www.computerworlduk.com/news/operating-systems/3242747/enterprises-planning-windows-7-upgrade-in-2011>.

Buller, J., Cerling, T., Enstall, C. & Ruiz, R. 2010. Mastering Microsoft Virtualization. Indianapolis: Wiley Publishing.

Easter, L. 2009. Getting Started with Application Virtualization white paper. Viitattu 15.2.2011. http://www.flexerasoftware.com/webdocuments/PDF/wp_as_GettingStartedwithAppV.pdf.

Georgieva, T. 2009. Disadvantages of Virtualization. Viitattu 1.4.2011. <http://www.suite101.com/content/disadvantages-of-virtualization-a170745>.

Hirsjärvi, S., Remes, P. & Sajavaara, P. 2009. Tutki ja kirjoita. 15. uud. p. Helsinki: Tammi.

HUS-Tietotekniikka. 2011. HUS-Tietotekniikan kotisivut. Viitattu 7.2.2011. <http://www.hus.fi/default.asp?path=1,28,824,26319>.

Madden, B. 2008. VMware buys Thinstall! What does this mean?. Viitattu 25.2.2011. <http://www.brianmadden.com/blogs/brianmadden/archive/2008/01/15/vmware-buys-thinstall-what-does-this-mean.aspx>.

Mangan, T. 2009. Microsoft App-V 4.5 Client in Stand-Alone Mode. Viitattu 27.2.2011. http://tmurgent.com/WhitePapers%5CMicrosoft_AppV_Stand-Alone.pdf.

Microsoft. 2006. Microsoft Completes Acquisition of Softricity. Viitattu 20.2.2011. <http://www.microsoft.com/presspass/press/2006/jul06/07-17SoftricityPR.msp>.

Microsoft. 2010a. Infrastructure Planning and Design – Microsoft Application Virtualization 4.6. Viitattu 20.2.2011. <http://go.microsoft.com/fwlink/?LinkId=160978>.

Microsoft. 2010b. Microsoft Application Virtualization 4.6 Sequencing Guide. Viitattu 27.2.2011. http://download.microsoft.com/download/F/7/8/F784A197-73BE-48FF-83DA-4102C05A6D44/APP-V/App-46_Sequencing_Guide_Final.docx.

Microsoft. 2010c. Virtual Application Management with Microsoft Application Virtualization 4.5/4.6 and System Center Configuration Manager 2007 R2. Viitattu 27.2.2011. http://download.microsoft.com/download/f/7/8/f784a197-73be-48ff-83da-4102c05a6d44/APP-V/App-V_and_ConfigMgr_Whitepaper_Final.docx.

Microsoft. 2011. Centralized Management & Scalable Infrastructure. Viitattu 26.2.2011. <http://www.microsoft.com/systemcenter/appv/infrastructure.mspx>.

Oksanen, M. 2010. Sovellusvirtualisointi. Oulun yliopiston tietohallinto. Viitattu 20.2.2011. <http://www.oulu.fi/tietohallinto/henkilokunnalle/uutiset/2010/1.html#thinapp>.

Pietroforte, M. 2008. The disadvantages of application virtualization. Viitattu 22.2.2011. <http://4sysops.com/archives/the-disadvantages-of-application-virtualization>.

Ruest, D & Ruest, N. 2009. Virtualization: A Beginner's Guide. McGraw-Hill. Viitattu 10.2.2011. <http://www.jamk.fi/kirjasto>, Nelli-portaali, Ebrary.

Spruijt, R. 2010. Application Virtualization Smackdown v3. Viitattu 11.2.2011. <http://www.virtuall.nl/download-document/application-virtualization-smackdown>.

VMware. 2009. VMware ThinApp Deployment Guide. Viitattu 26.2.2011. http://www.vmware.com/files/pdf/VMware_ThinApp_Deployment_Guide.pdf.

VMware. 2010a. VMware ThinApp Reviewer's Guide. Viitattu 24.2.2011. <http://www.vmware.com/files/pdf/VMware-ThinApp-Reviewers-Guide.pdf>.

VMware. 2010b. ThinApp 4.6 User's Guide. Viitattu 25.2.2011. http://www.vmware.com/pdf/thinapp46_manual.pdf.

VMware. 2010c. VMware Thinapp 4.6 – What's New?. Viitattu 24.2.2011. <http://blogs.vmware.com/thinapp/2010/08/vmware-thinapp-46-whats-new.html>.

VMware. 2011a. History of Virtualization. Viitattu 7.2.2011. <http://vmware.com/virtualization/history.html>.

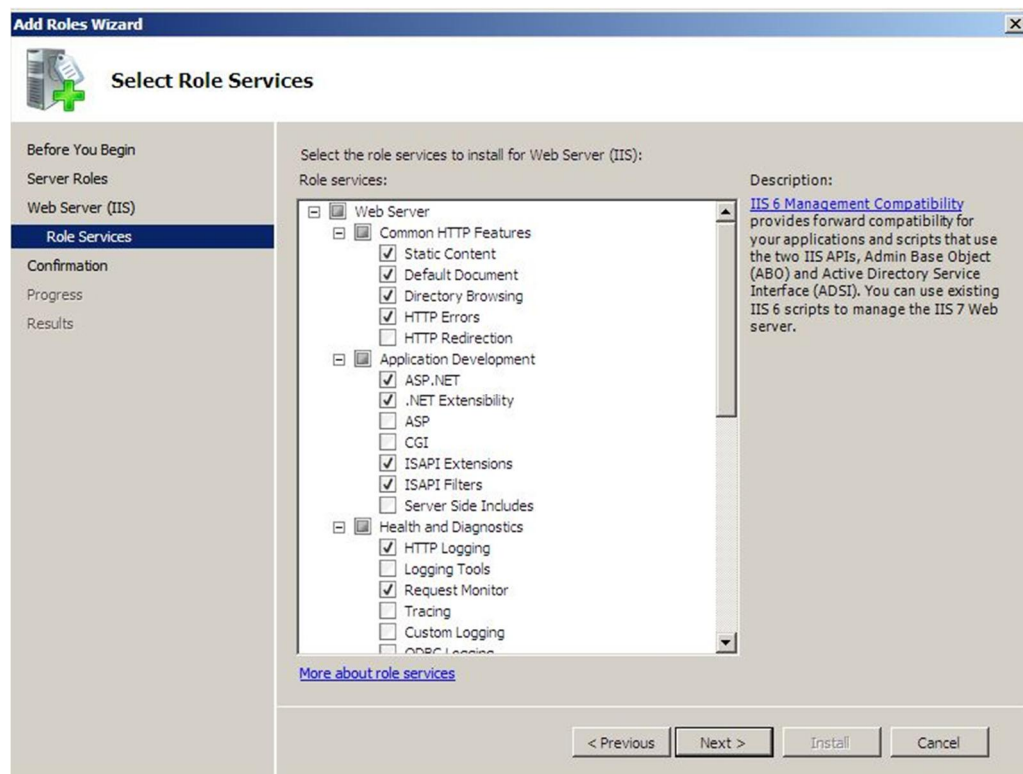
VMware. 2011b. Windows 7 Migration. Viitattu 15.2.2011. <http://www.vmware.com/solutions/desktop/windows7-migration.html>.

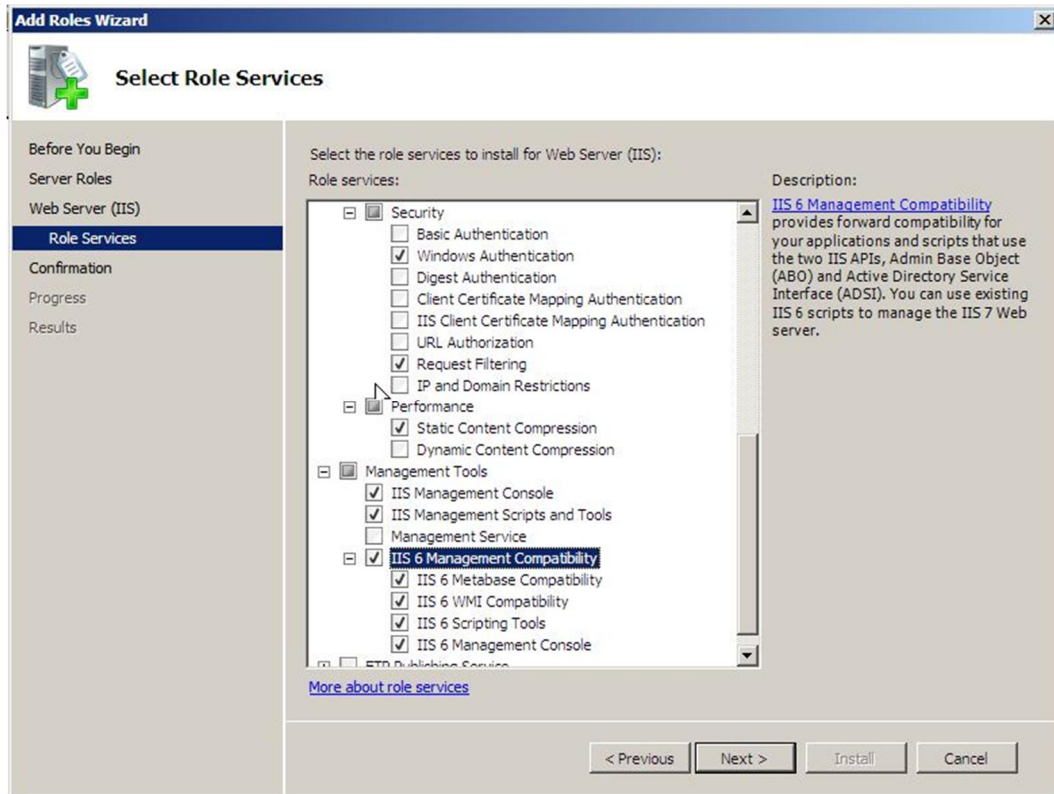
Wikipedia. 2011. Application Virtualization. Viitattu 22.2.2011. http://en.wikipedia.org/wiki/Application_virtualization.

LIITTEET

Liite 1. IIS:n asennus

IIS-rooli asennetaan Server Managerista Add Roles-kohdasta. IIS:n asennuksessa pitää aktivoida oletusasetuksien lisäksi ASP.NET-, Windows Authentication-, IIS Management Scripts and Tools- sekä IIS 6 Management Compatibility-palvelut.





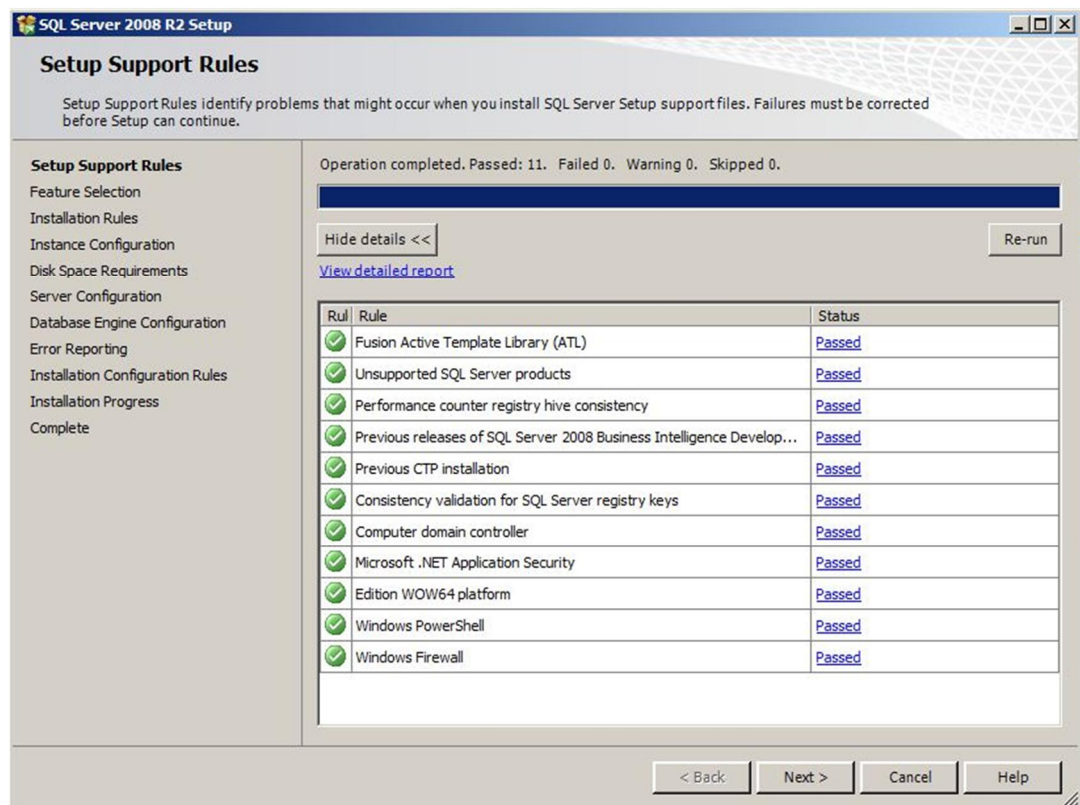
Liite 2. SQL-palvelimen asennus

SQL-palvelin vaatii asentuaakseen powershellin. Powershellin saa asennettua Server Managerissa Add Features -kohdasta. SQL-palvelimen asennus on nähtävissä alla.

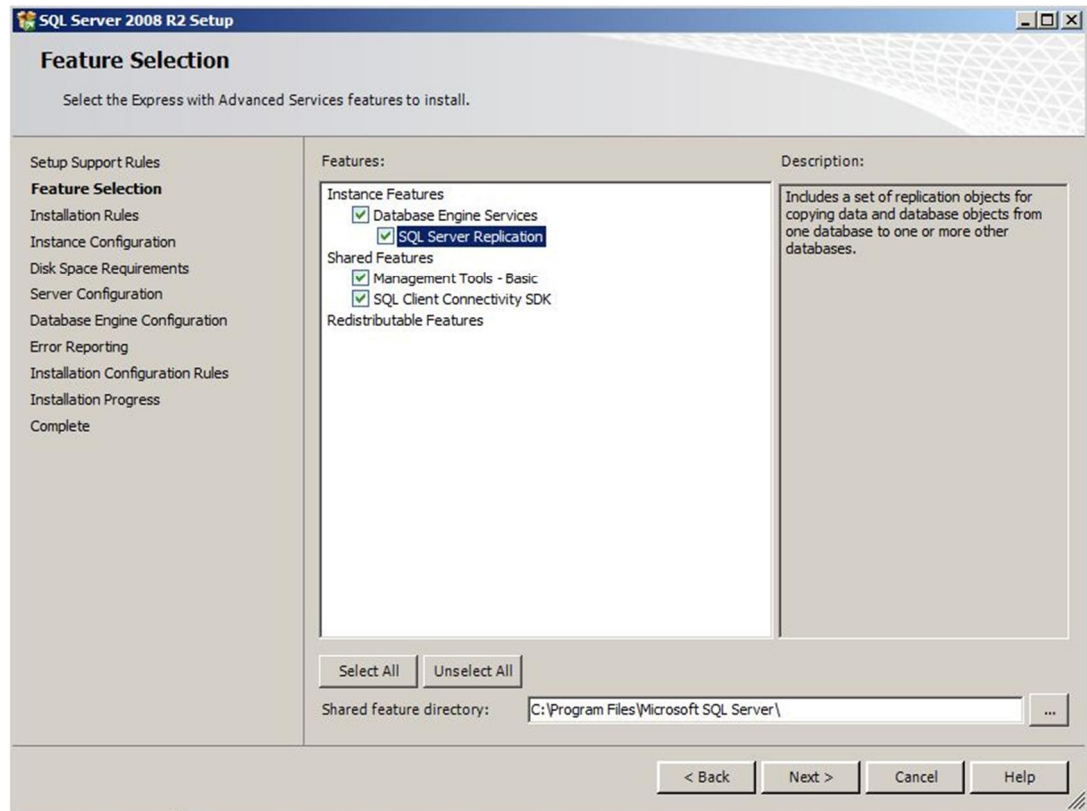
Valitaan New Installation or add features to an existing installation.



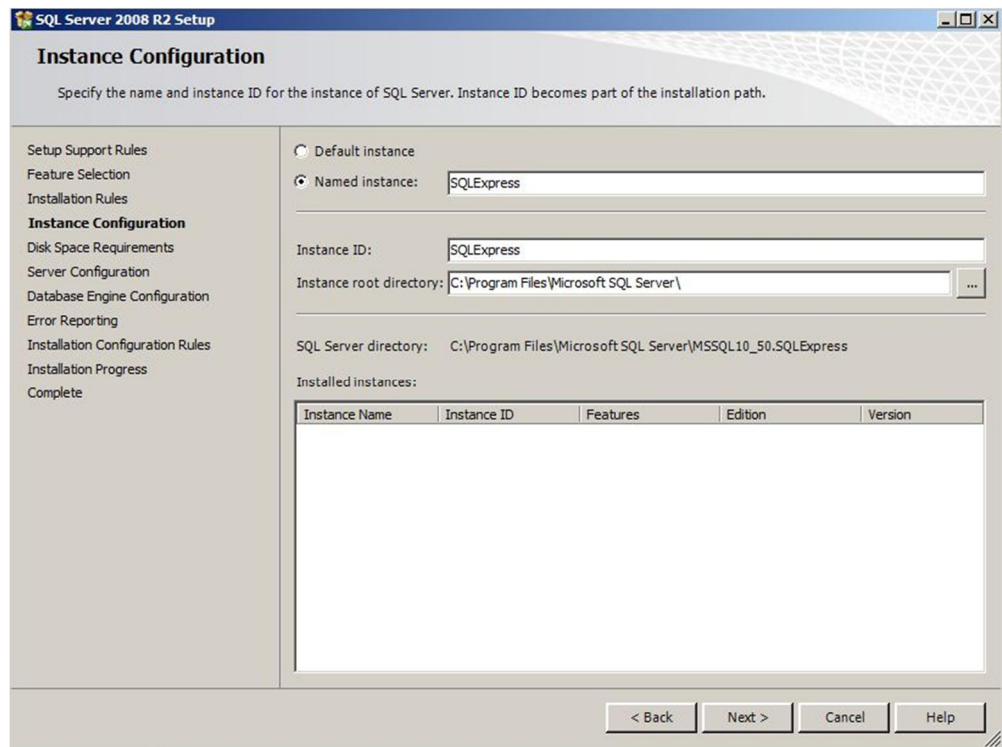
Asennusohjelma tarkistaa järjestelmän soveltuvuuden useilla eri testeillä.



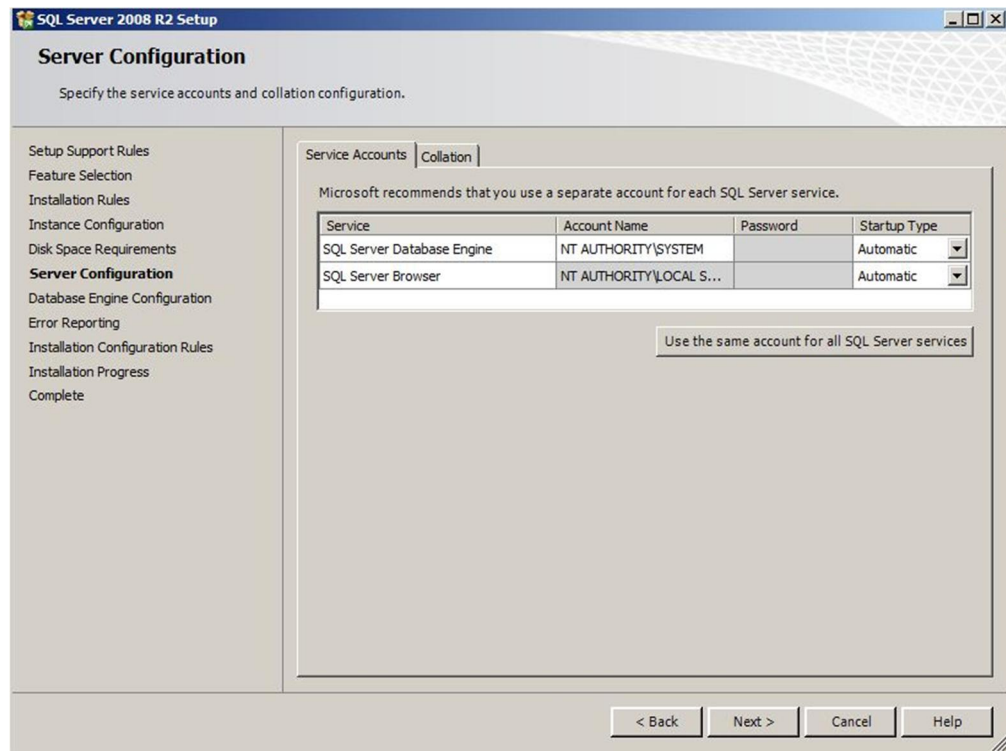
SQL Server Replication sekä SQL Client Connectivity eivät ole pakollisia asennettavina ominaisuuksina.



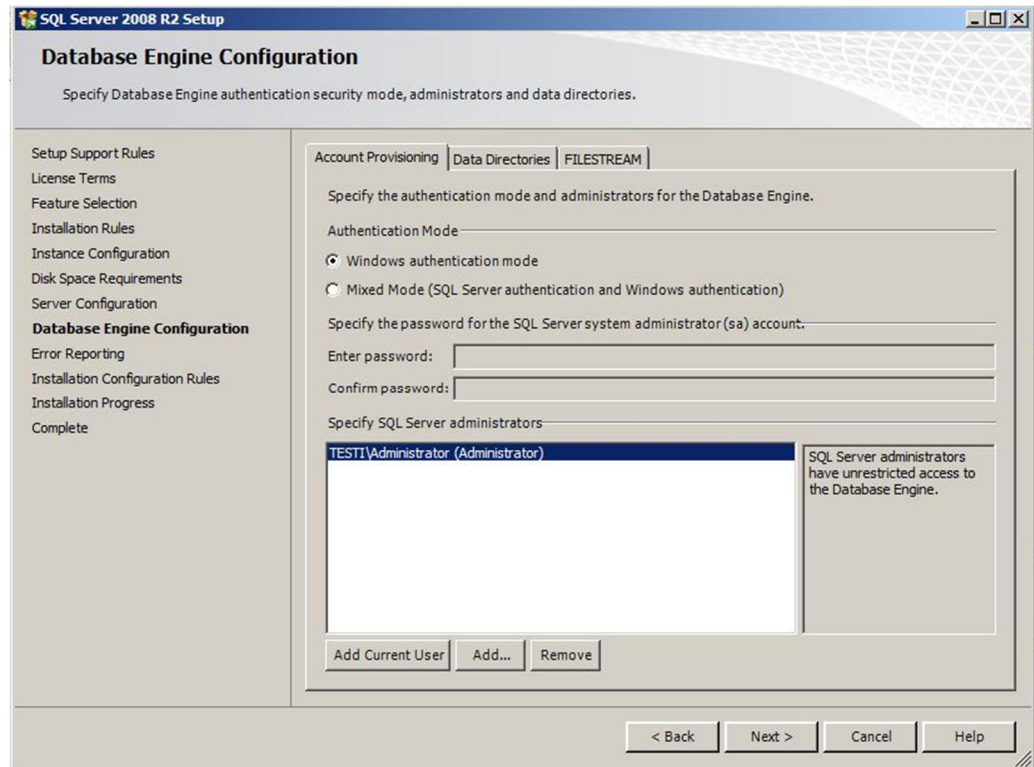
Luodaan uusi instanssi SQLExpress.



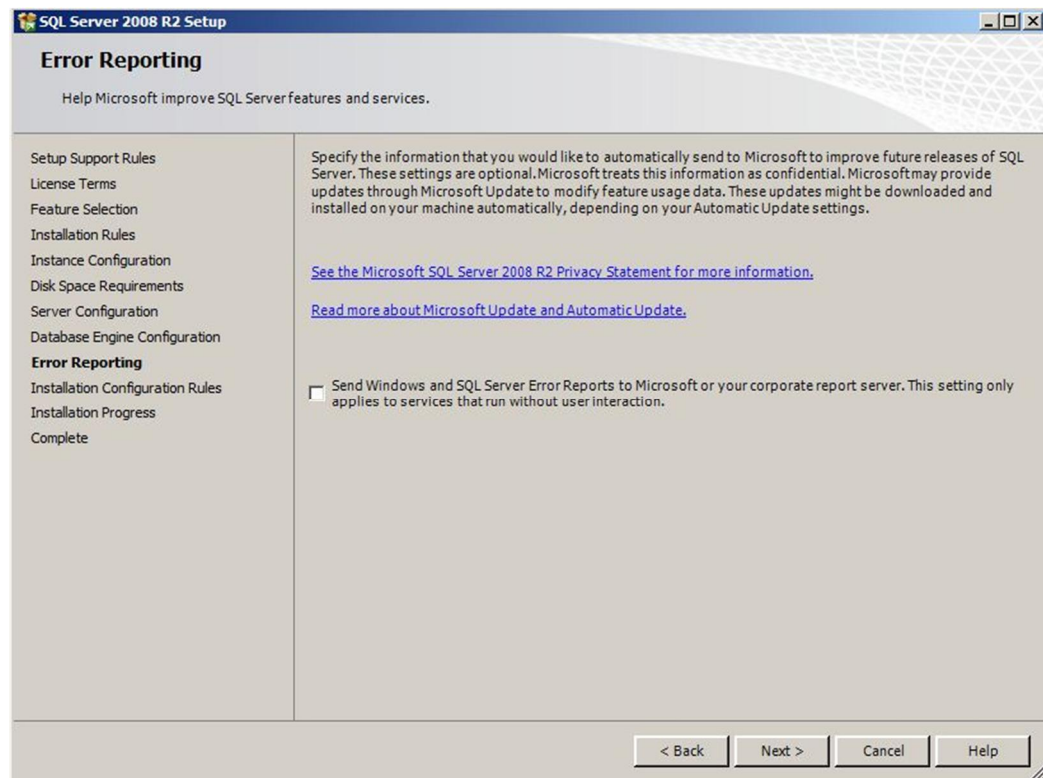
Vaihdetaan SQL Server Database Engine -palvelun Account Name -kohtaan NT Authority\SYSTEM sekä muutetaan SQL Server Browser:n Startup Type arvoon Automatic.



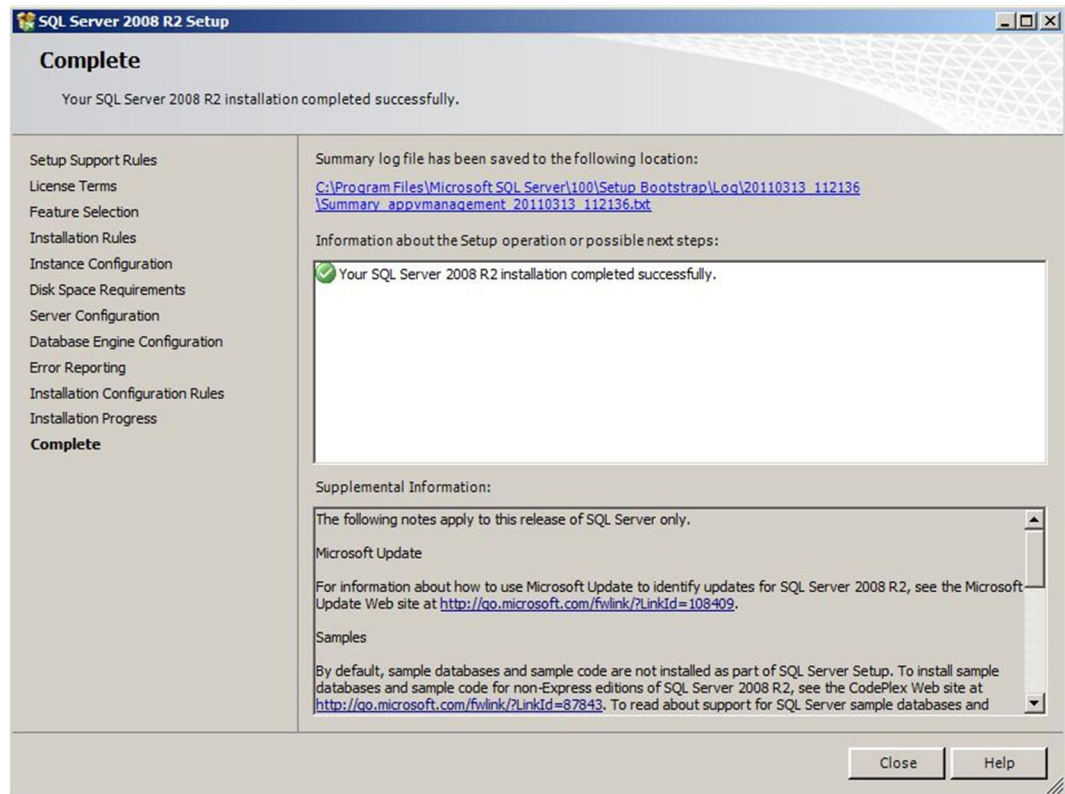
Käytetään Windows authentication mode -asetusta autentikointiin.



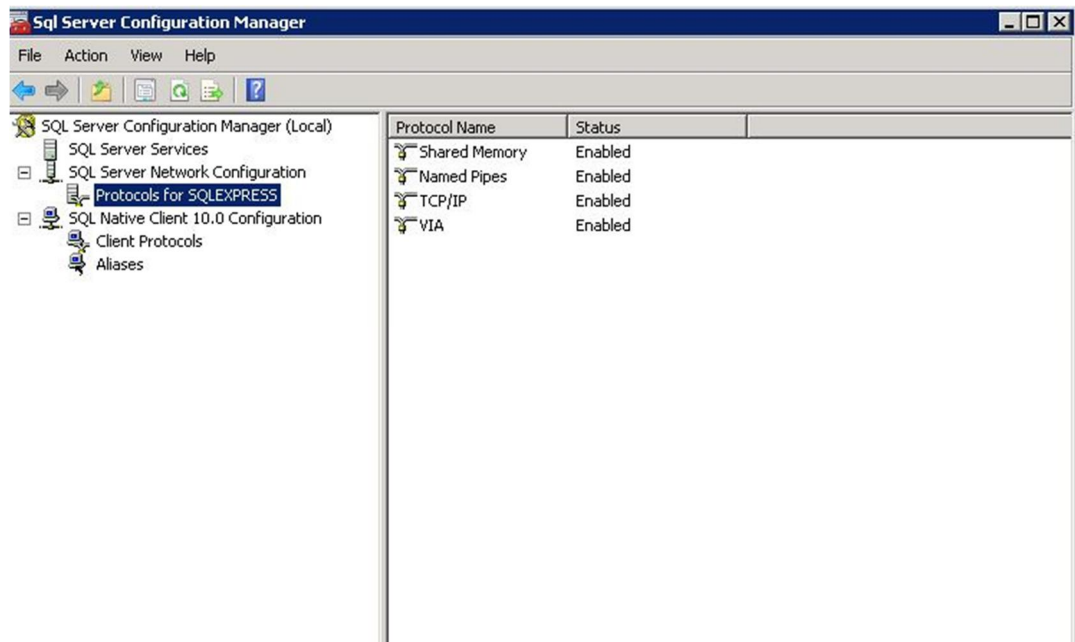
Virheiden raportointia Microsoftille ei tarvitse käyttää.



Sovellusvelho ilmoittaa, että SQL Server 2008 R2 Expressin asennus on suoritettu onnistuneesti.



Asennuksen jälkeen avataan vielä SQL Server Configuration Manager ja aktivoidaan käytettävät protokollat alla olevan kuvan mukaisesti

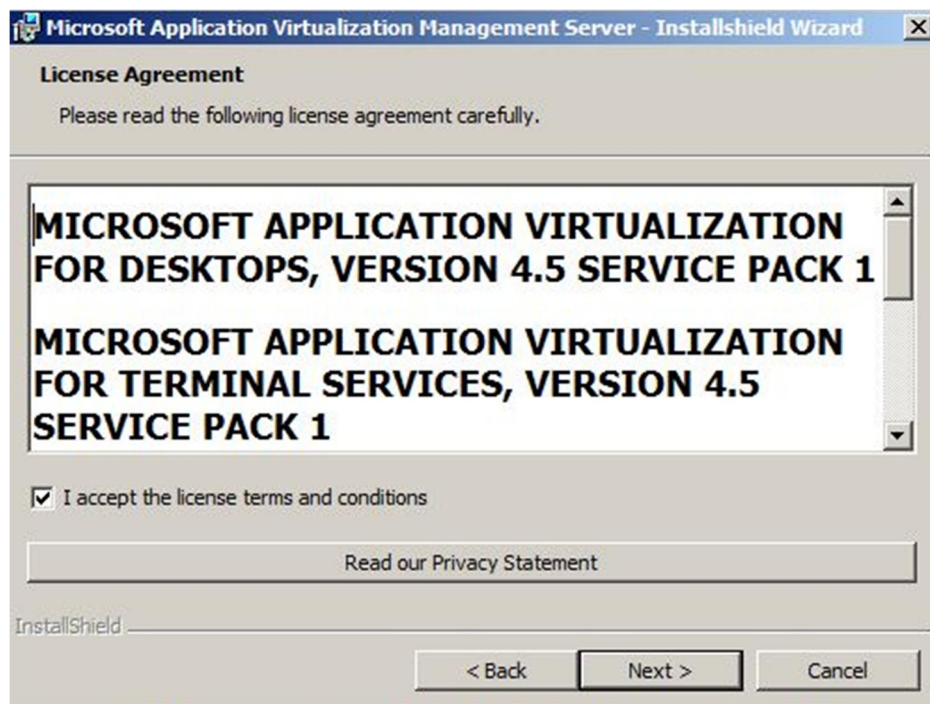


Liite 3. Microsoft Application Virtualization Management Server- in asennus

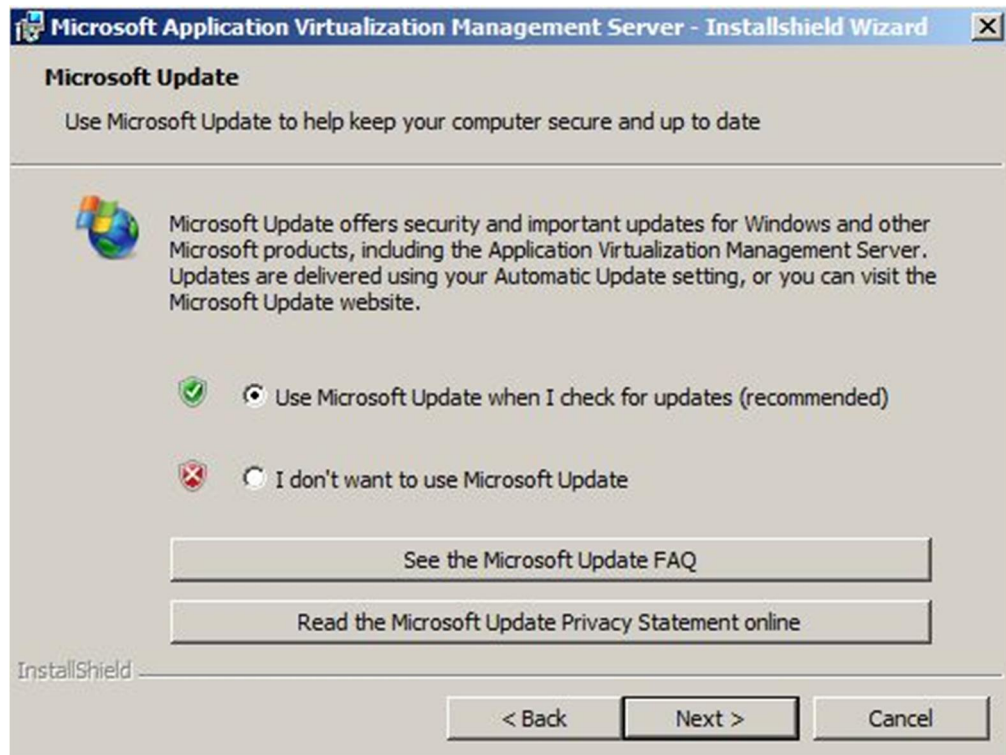
Käynnistetään Management-palvelimen asennus.



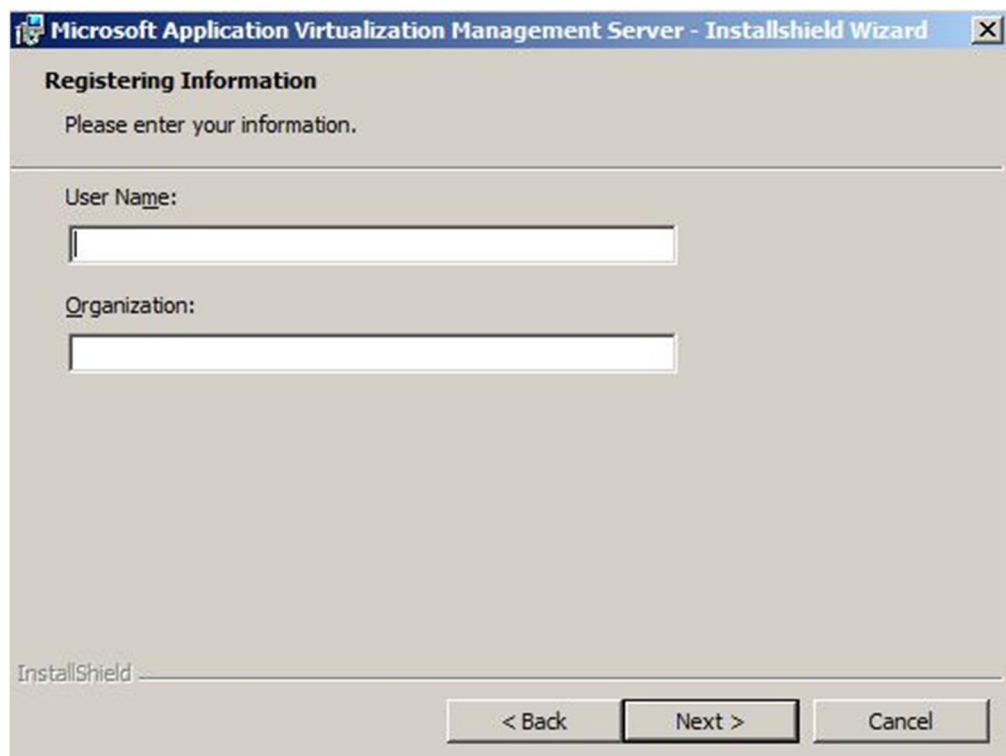
Hyväksytään lisenssiehdot.



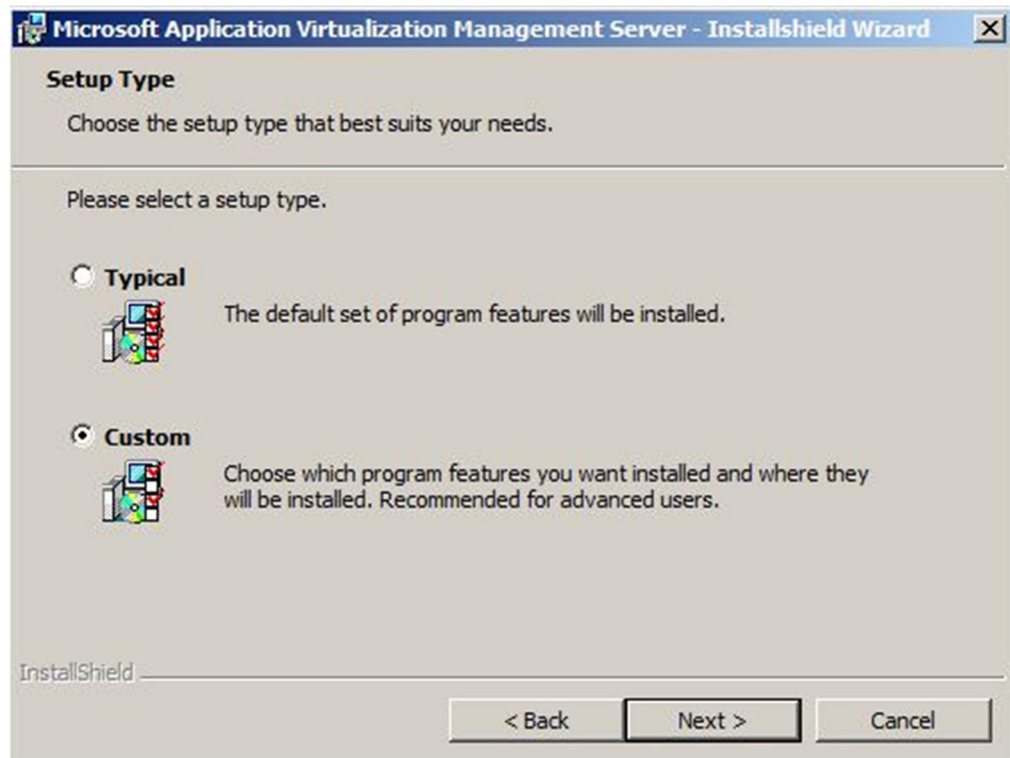
Käytetään Microsoftin päivitystoimintoa.



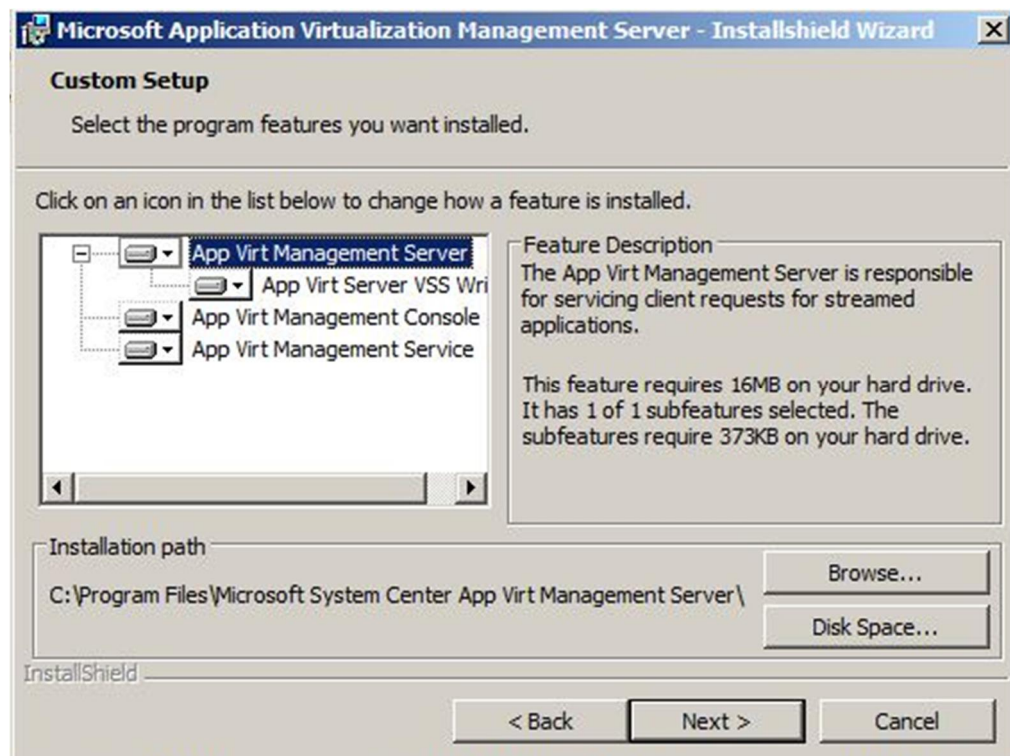
Täytetään rekisteröintitiedot.



Valitaan Custom-vaihtoehto asetusten muuttamiseksi.



Asennetaan Management Server, Console sekä Service.



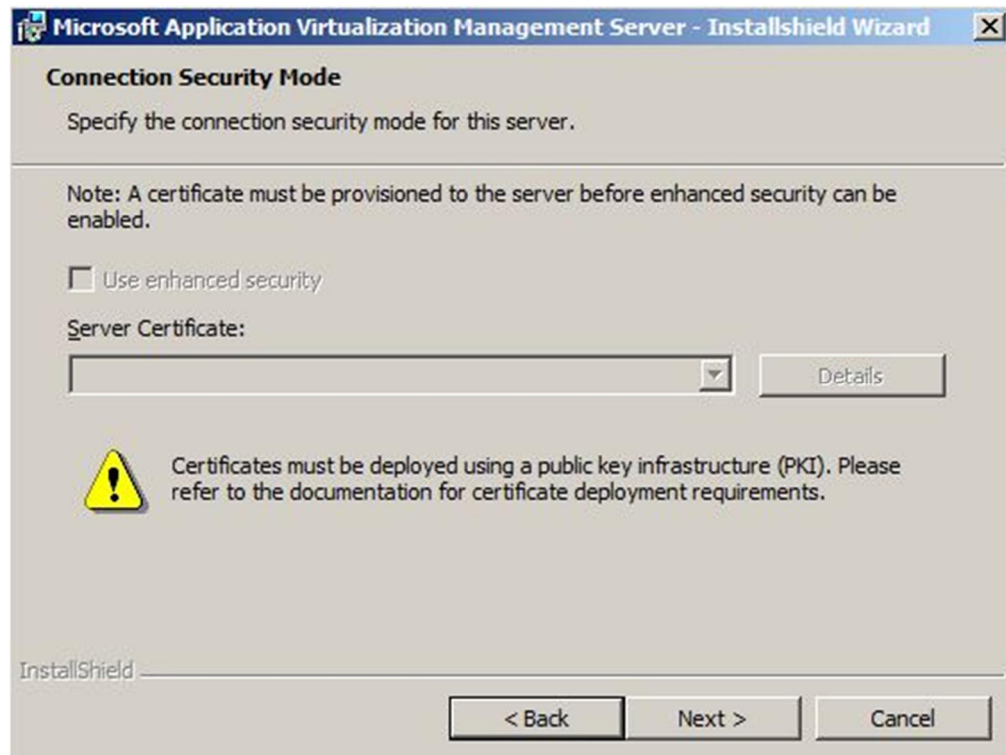
Yhdistetään luotuun SQL-palvelimeen, jos palvelin ei näy listassa, se pitää kirjoittaa käsin alempaan laatikkoon.

The screenshot shows the 'Configuration Database' step of the 'Microsoft Application Virtualization Management Server - InstallShield Wizard'. The window title is 'Microsoft Application Virtualization Management Server - InstallShield Wizard'. The main heading is 'Configuration Database'. Below it, the text says 'Please provide the following information about your database server.' The next instruction is 'Select the database server to be used for the Application Virtualization Management Server configuration information from the following drop down list of detected database servers.' There is a 'Server name:' label followed by a dropdown menu showing 'APPVMANAGEMENT\SQLEXPRESS'. Below this, the text says 'If your database server does not appear in the drop down list, provide the server name to connect below.' There is a checkbox labeled 'Use the following host name to access the database server.' which is currently unchecked. Below the checkbox are two input fields: 'Server Name:' and 'Port:'. The 'Port:' field contains the value '1433'. At the bottom left, the text 'InstallShield' is visible. At the bottom right, there are three buttons: '< Back', 'Next >', and 'Cancel'.

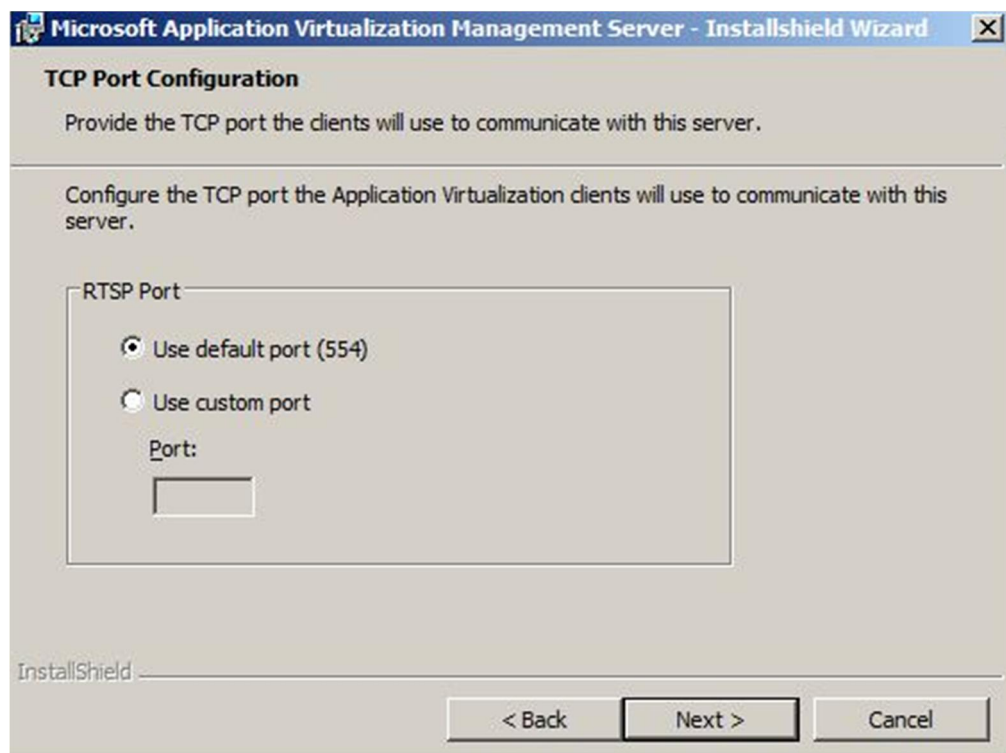
Luodaan uusi tietokanta APPVIRT.

The screenshot shows the 'Configuration Database' step of the 'Microsoft Application Virtualization Management Server - InstallShield Wizard'. The window title is 'Microsoft Application Virtualization Management Server - InstallShield Wizard'. The main heading is 'Configuration Database'. Below it, the text says 'Please provide the following information about your database.' The next instruction is 'Choose one of the following options:'. There are two radio button options: 'Use an existing database' and 'Create a new database'. The 'Create a new database' option is selected. To the right of the 'Create a new database' option is a text input field containing the value 'APPVIRT'. Below this, the text says 'To specify a different location other than the server default for the new database, enter the location path below. The path must be an existing, valid directory'. There is a checkbox labeled 'Use the following location when creating the database:' which is currently unchecked. Below the checkbox is a 'Path:' label followed by an empty text input field. At the bottom left, the text 'InstallShield' is visible. At the bottom right, there are three buttons: '< Back', 'Next >', and 'Cancel'.

Ei käytetä sertifikaattia tässä tapauksessa.



Käytetään RTSP:n oletusporttia 554



Määritetään admin-ryhmä, jolla Management Consolea voidaan käyttää hallintapalvelimen hallitsemiseen.

Microsoft Application Virtualization Management Server - Installshield Wizard

Administrator Group

Please specify the administrator group for the server.

Application Virtualization Management Servers are configured using the Application Virtualization Management Console. Specify the security group that is authorized to manage your server.

To search the list of groups, enter the first few characters of the desired group name, then click Next.

Group Name:

InstallShield

< Back Next > Cancel

Määritetään Default Provider-ryhmä.

Microsoft Application Virtualization Management Server - Installshield Wizard

Default Provider Group

Please specify the default user group for the server.

By default, Application Virtualization Management Servers are setup to authenticate users using Windows Authentication. Specify the security group that is authorized to use Application Virtualization Management Servers.

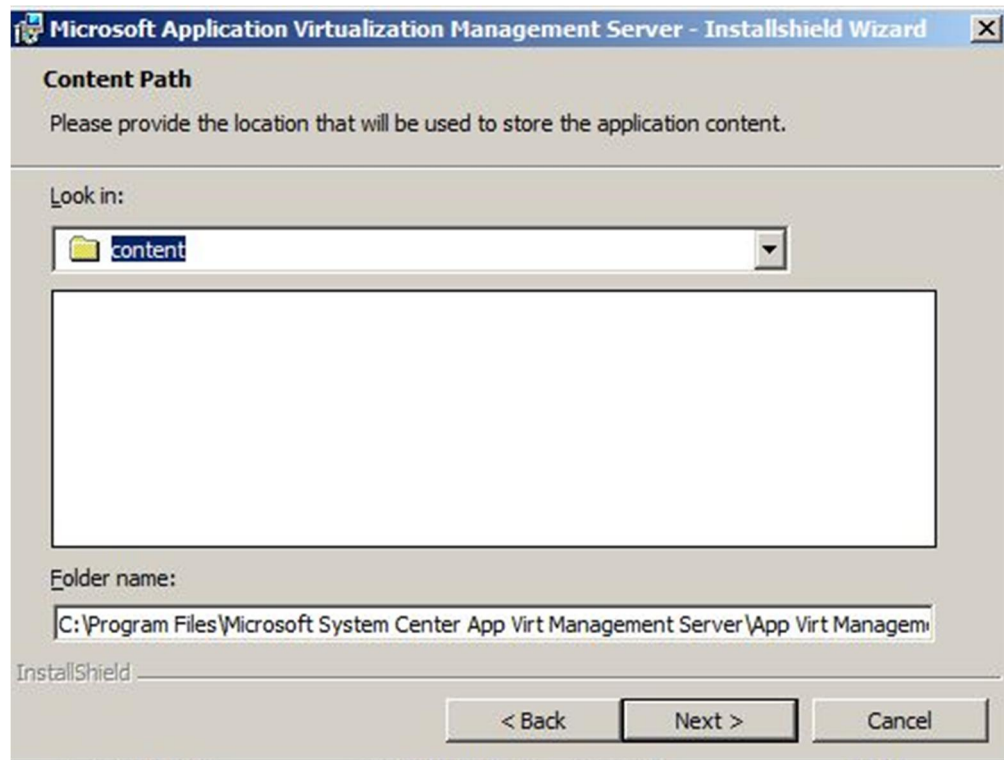
To search the list of groups, enter the first few characters of the desired group name, then click Next.

Group Name:

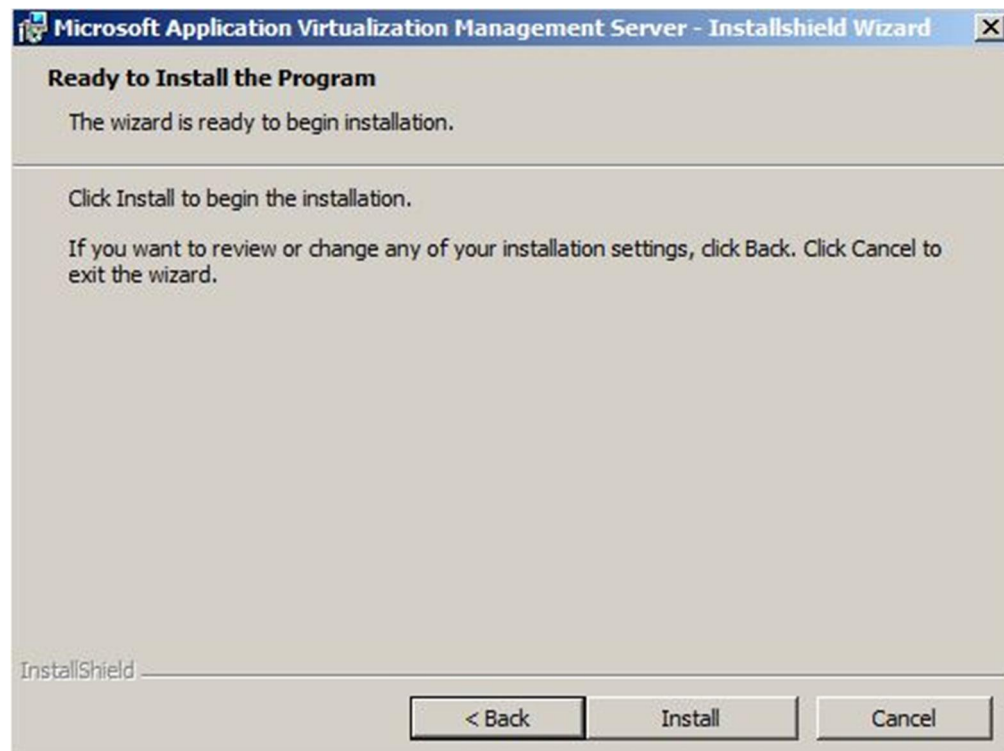
InstallShield

< Back Next > Cancel

Määritetään content-kansion sijainti virtualisoituja sovelluksia varten.

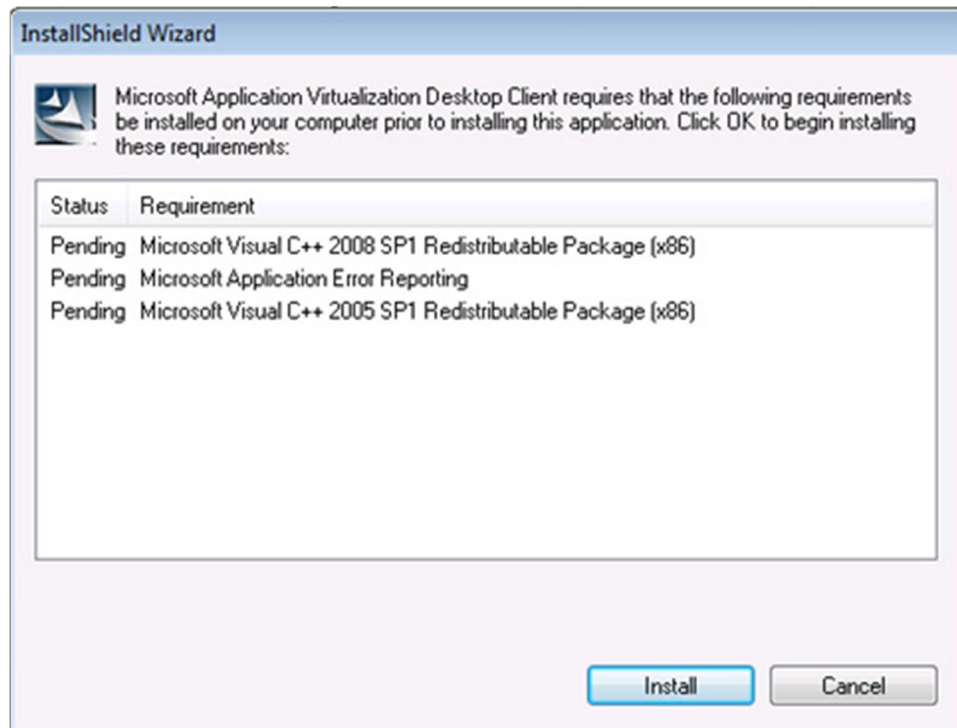


Suoritetaan asennus loppuun.

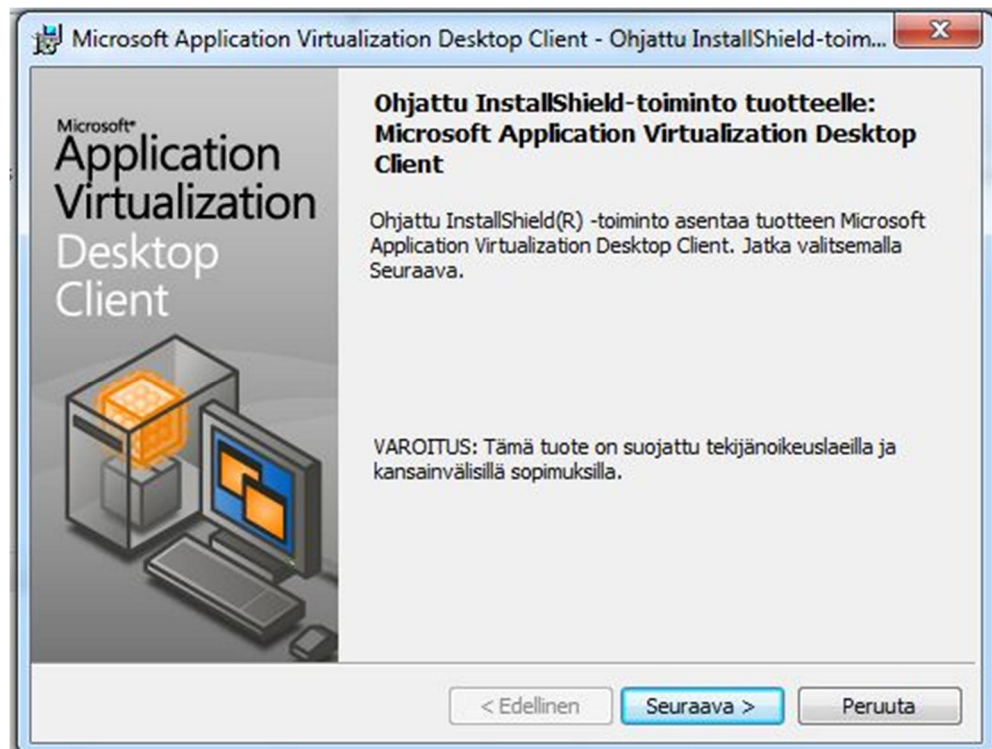


Liite 4. Asiakasohjelman asennus

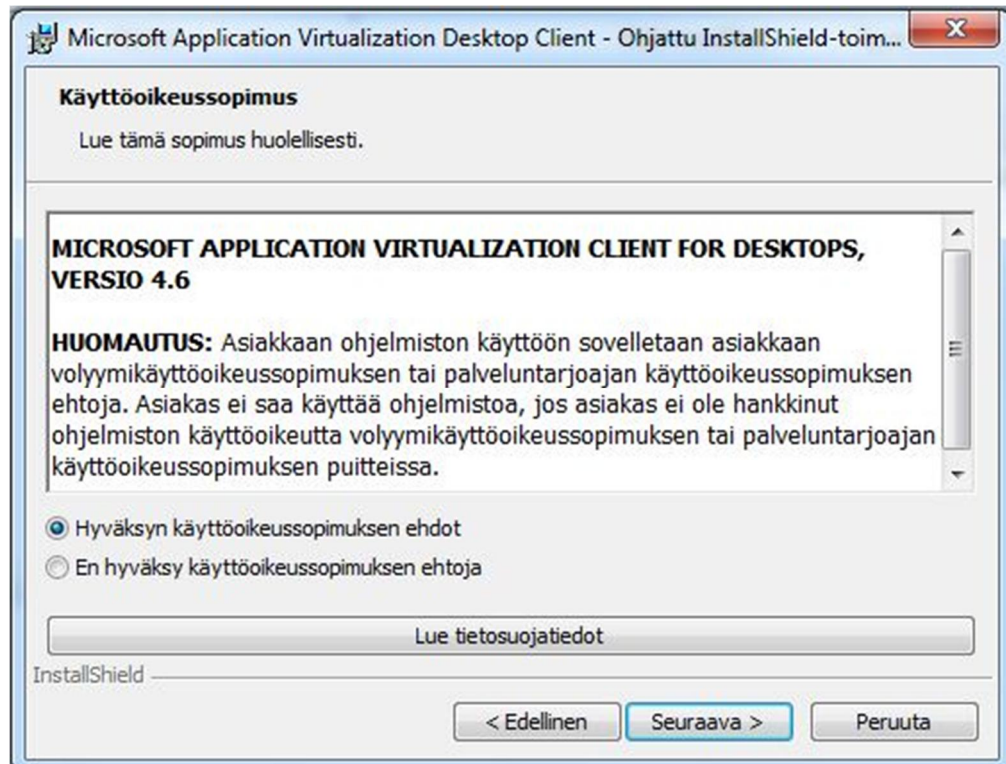
Asiakasohjelma vaatii alla olevat komponentit asentuakseen.



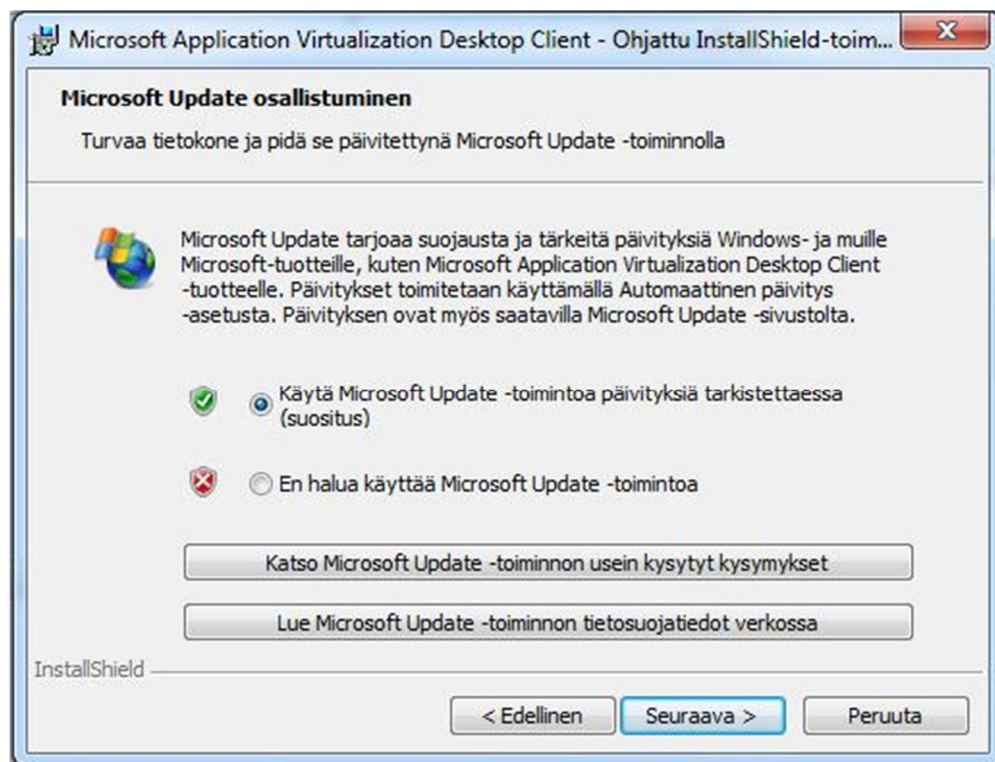
Aloitetaan asiakasohjelman asennus



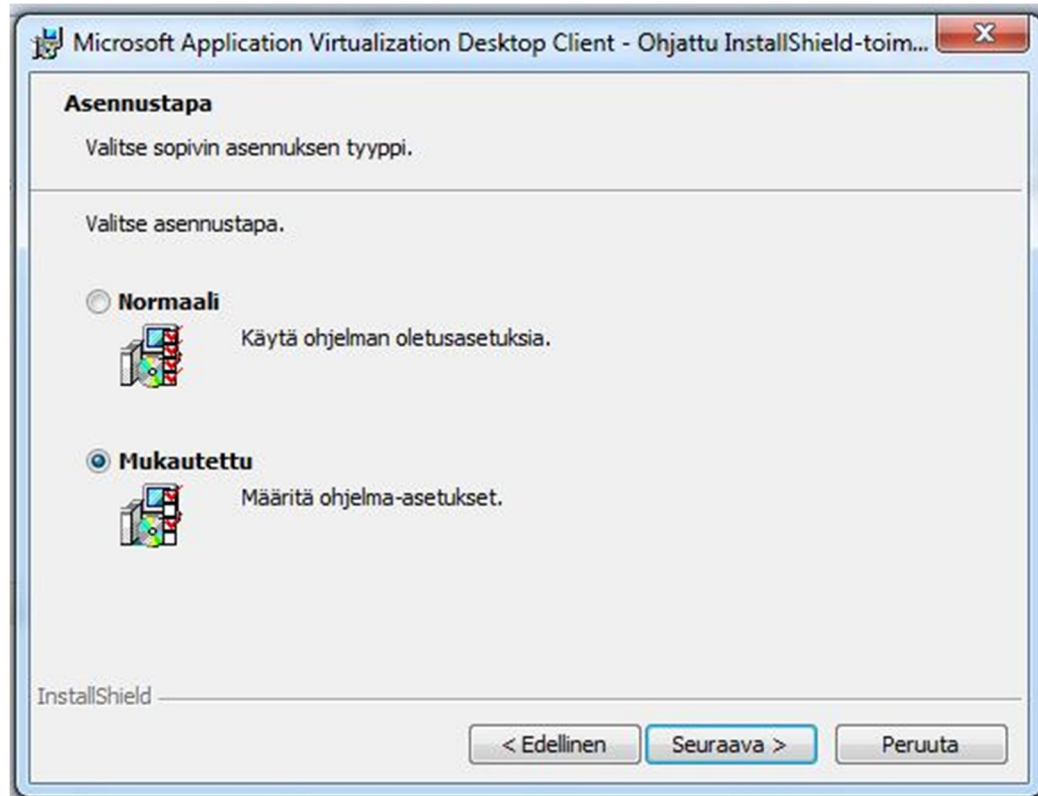
Hyväksytään käyttöehdot



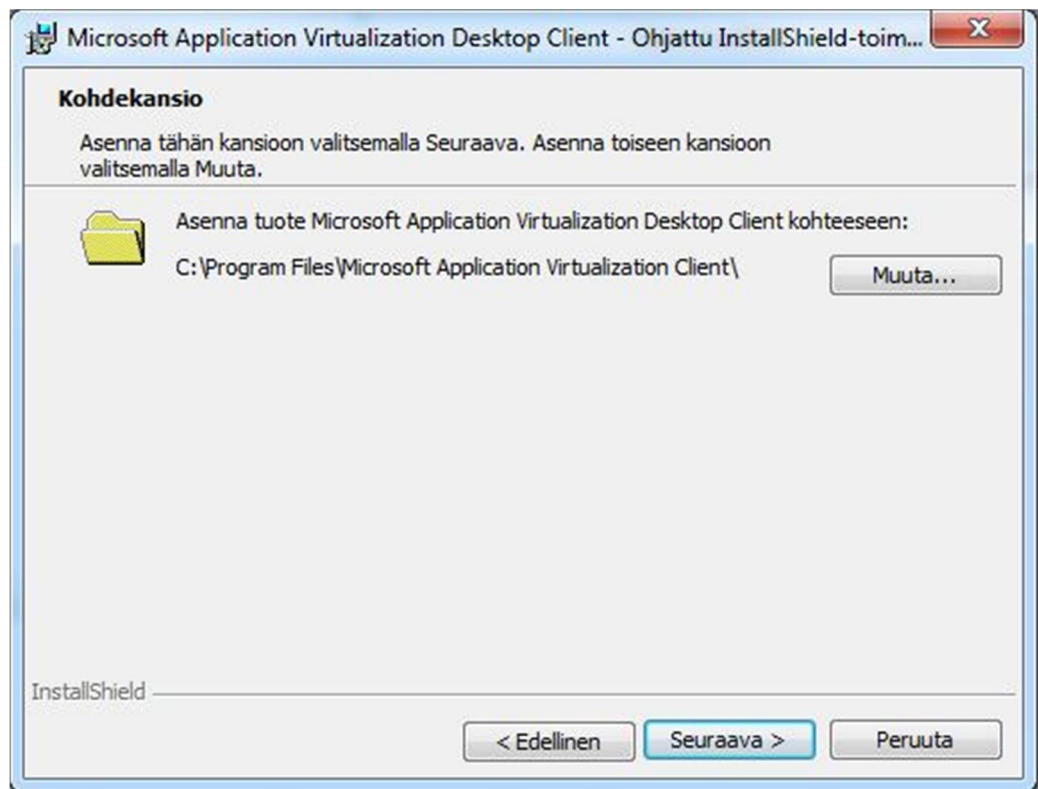
Käytetään Microsoft Updatea päivitysten tarkistamiseen.



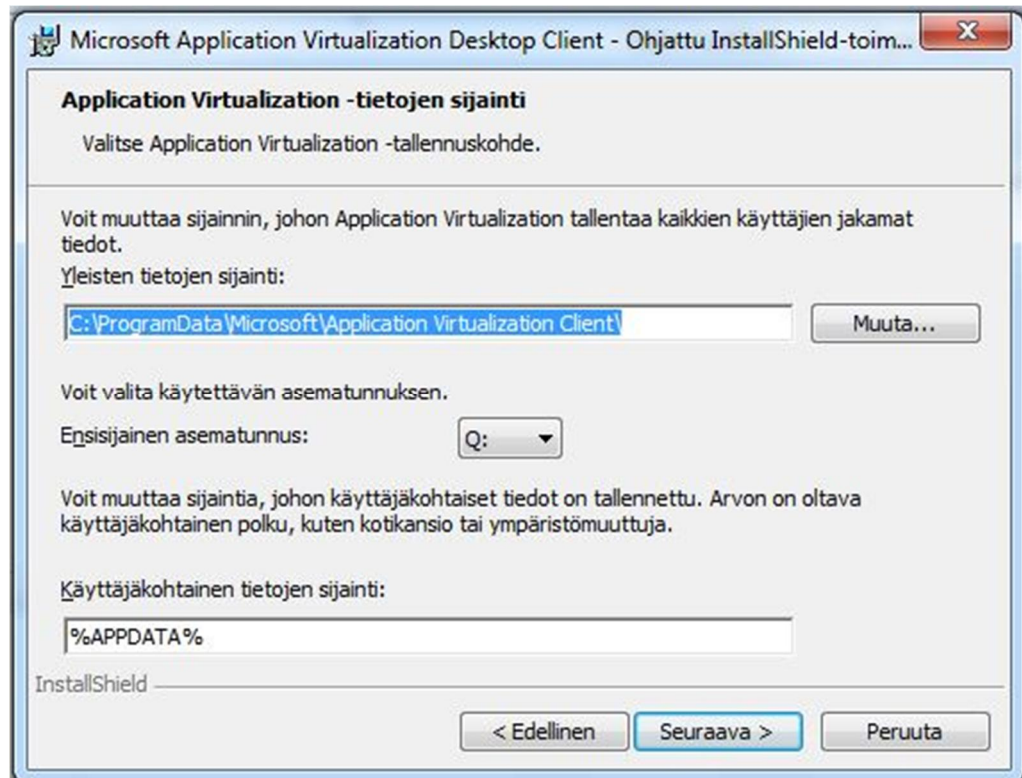
Valitaan mukautettu asennus asetuksien muuttamiseksi.



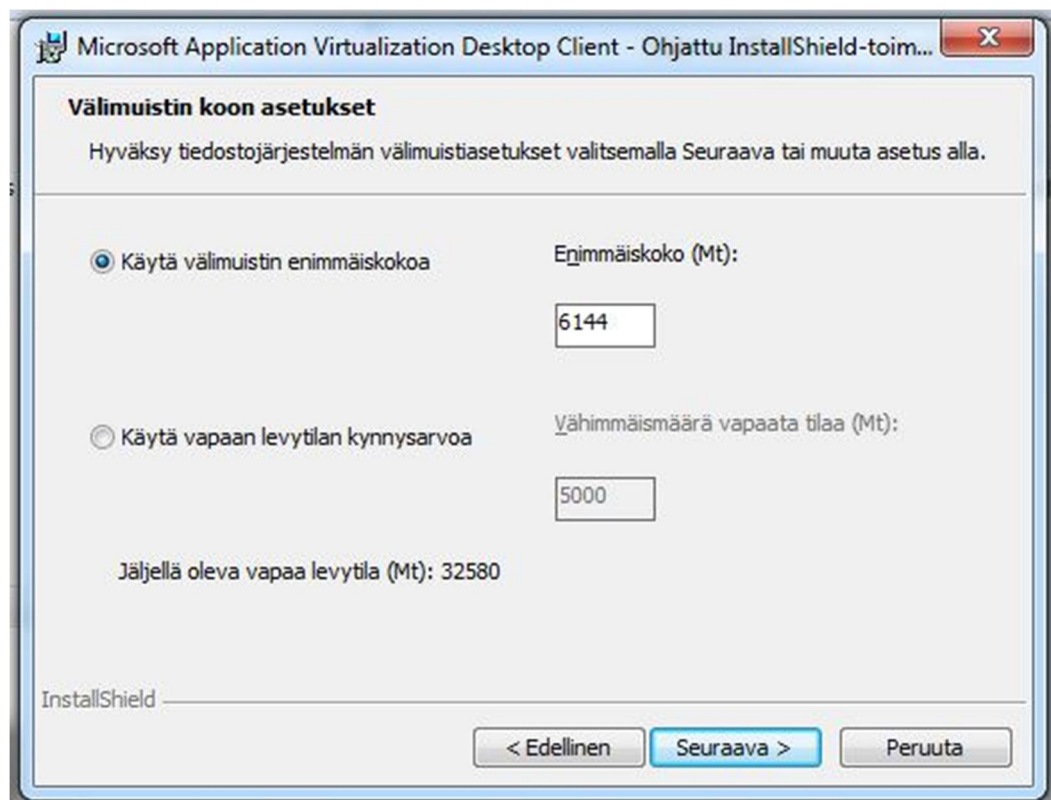
Jätetään kohdekansioiksi asennusvelhon ehdottama sijainti.



Varmistetaan, että ensisijainen asematunnus on Q ja jätetään oletusasetukset.



Käytetään välimuistin enimmäiskoko -asetusta. Jos tarvitaan suurempi välimuisti, voidaan käyttää käytä vapaan levytilan kynnyсарvo -asetusta.



Asetuksia voi muuttaa halutulla tavalla.

Suoritusajapaketin käytännön määrittely

Aseta parametrit, jotka määrittävät kuinka Application Virtualization Client käyttäytyy suoritusajapaketina.

Application Source Root

Sovelluksen käyttöoikeudet

- ☒ Vaadi käyttäjän valtuutus myös välimuistissa
- ☐ Salli virtauttaminen tiedostosta

Sovelluksen automaattinen lataaminen

Automaattisen latauksen suorittamisajankohta:

- ☒ Käynnistyksen yhteydessä
- ☒ Sisäänkirjautumisen yhteydessä
- ☐ Päivityksen julkaisun yhteydessä

Automaattisella latauksella ladattavat sovellukset:

- ☐ Älä lataa sovelluksia automaattisesti.
- ☒ Lataa aiemmin käytetyt sovellukset automaattisesti.
- ☐ Lataa kaikki sovellukset automaattisesti.

InstallShield

< Edellinen Seuraava > Peruuta

Määritetään hallintapalvelimen näyttönimi ja isäntänimi. Isäntänimi on aiemmin luotu palvelin. Näyttönimi voi olla mitä vaan.

Julkaisupalvelin

Määritä julkaisupalvelin.

☒ Määritä julkaisupalvelin

Näyttönimi:

Management Server

Tyyppi:

Application Virtualization Server

Isäntänimi:

appvmanagement

Portti:

554

Polku:

/

☒ Muodosta automaattisesti yhteys tähän palvelimeen asetusten päivitystä varten kun käyttäjä kirjautuu sisään

InstallShield

< Edellinen Seuraava > Peruuta

Lopuksi valitaan Asenna sovelluksen asentamiseksi.

